

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4888729号  
(P4888729)

(45) 発行日 平成24年2月29日 (2012. 2. 29)

(24) 登録日 平成23年12月22日 (2011. 12. 22)

(51) Int. Cl.	F I
HO 4 N 1/413 (2006. 01)	HO 4 N 1/413 Z
GO 6 T 9/00 (2006. 01)	GO 6 T 9/00
HO 3 M 7/40 (2006. 01)	HO 3 M 7/40
HO 4 N 7/26 (2006. 01)	HO 4 N 7/13 Z

請求項の数 38 (全 80 頁)

(21) 出願番号	特願2007-524111 (P2007-524111)	(73) 特許権者	000002185
(86) (22) 出願日	平成18年11月17日 (2006. 11. 17)		ソニー株式会社
(86) 国際出願番号	PCT/JP2006/322953		東京都港区港南1丁目7番1号
(87) 国際公開番号	W02007/058296	(74) 代理人	100082131
(87) 国際公開日	平成19年5月24日 (2007. 5. 24)		弁理士 稲本 義雄
審査請求日	平成21年11月11日 (2009. 11. 11)	(72) 発明者	福原 隆浩
(31) 優先権主張番号	特願2005-334243 (P2005-334243)		東京都港区港南1丁目7番1号 ソニー株
(32) 優先日	平成17年11月18日 (2005. 11. 18)		式会社内
(33) 優先権主張国	日本国 (JP)	(72) 発明者	保坂 和寿
(31) 優先権主張番号	特願2006-7256 (P2006-7256)		東京都港区港南1丁目7番1号 ソニー株
(32) 優先日	平成18年1月16日 (2006. 1. 16)		式会社内
(33) 優先権主張国	日本国 (JP)	(72) 発明者	安藤 勝俊
			東京都港区港南1丁目7番1号 ソニー株
			式会社内

最終頁に続く

(54) 【発明の名称】 符号化装置および方法、並びに、復号装置および方法

(57) 【特許請求の範囲】

【請求項 1】

係数データを符号化する符号化装置において、

前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号を生成する最大有効桁数生成手段と、

前記係数データにより表わされる複数の数値の絶対値を示す符号を生成する絶対値生成手段と、

前記係数データにより表わされる複数の数値の正負を示す符号を生成する符号生成手段と、

前記最大有効桁数生成手段により生成された最大有効桁数を示す符号と、前記絶対値生成手段により生成された絶対値を示す符号と、前記符号生成手段により生成された正負を示す符号とを連結して、符号化データを生成する符号連結手段と

を備える符号化装置。

【請求項 2】

画像データに対して直交変換を行って、前記係数データを生成する直交変換手段と、

前記直交変換手段により生成された係数データを記憶する記憶手段と、

前記記憶手段により記憶された前記係数データを、前記係数データに対して逆直交変換を行う順に予め並び替える係数並び替え手段と

をさらに備える請求項 1 に記載の符号化装置。

**【請求項 3】**

画像データに対して、少なくとも最低域成分のサブバンドの 1 ライン分の係数データを生成するのに必要なライン数分の画像データを含むラインブロック毎にフィルタ処理を行い、周波数帯域毎に分解された前記係数データからなる複数のサブバンドを生成するフィルタ手段と、

前記フィルタ手段により生成された係数データをラインブロック毎に記憶する記憶手段と、

前記記憶手段により記憶された前記係数データを、ラインブロック毎に、周波数帯域に分解された複数のサブバンドの係数データを合成して画像データを生成する合成処理を実行する順に予め並び替える係数並び替え手段と

をさらに備える請求項 1 に記載の符号化装置。

**【請求項 4】**

前記係数並び替え手段は、前記係数データを、ラインブロック毎に、低域成分から高域成分の順に並び替える

請求項 1 に記載の符号化装置。

**【請求項 5】**

前記最大有効桁数生成手段は、前記最大有効桁数を示す符号として、最大有効桁数が変化したか否かを示す符号を生成する

請求項 1 乃至請求項 3 のいずれかに記載の符号化装置。

**【請求項 6】**

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数が増加したかまたは減少したかを示す符号を生成する

請求項 5 に記載の符号化装置。

**【請求項 7】**

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数の変化量を示す符号を生成する

請求項 6 に記載の符号化装置。

**【請求項 8】**

前記最大有効桁数生成手段は、連続して並べられた前記係数データの前記最大有効桁数を示す符号を生成し、前記係数データにさらに連続して並べられた前記係数データの最大有効桁数を示す符号を生成する処理を繰り返し行う

請求項 1 乃至請求項 3 のいずれかに記載の符号化装置。

**【請求項 9】**

前記絶対値生成手段は、前記係数データにより表わされる複数の数値の絶対値を示す符号として、前記係数データが表わす数値の最下位の桁から前記最大有効桁までの値を示す符号を生成する

請求項 1 乃至請求項 3 のいずれかに記載の符号化装置。

**【請求項 10】**

前記絶対値生成手段は、前記係数データにより表わされる複数の数値の絶対値を示す符号を並列に生成する

請求項 1 乃至請求項 3 のいずれかに記載の符号化装置。

**【請求項 11】**

前記係数データにより表される複数の数値が全て 0 であるか否かを示す符号を生成するゼロ符号生成手段をさらに備え、

前記最大有効桁数生成手段は、前記ゼロ符号生成手段により生成される 0 であるか否かを示す符号に示される値に従って、前記最大有効桁数を示す符号を生成する

請求項 1 乃至請求項 3 のいずれかに記載の符号化装置。

**【請求項 12】**

前記絶対値生成手段は、前記最大有効桁数生成手段により生成される数値の絶対値を示す符号に示される値に従って、数値の絶対値を示す符号を生成し、

10

20

30

40

50

前記符号生成手段は、前記絶対値生成手段により生成される数値の絶対値を示す符号に示される値に従って、数値の正負を示す符号を生成する

請求項 1 1 に記載の符号化装置。

【請求項 1 3】

係数データを符号化する符号化方法において、

前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号を生成し、

前記係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号を生成し、

前記係数データにより表わされるそれぞれの数値の正負を示す複数の符号を生成し、

生成された最大有効桁数を示す符号と、生成された絶対値を示す符号と、生成された正負を示す符号とを連結して、符号化データを生成する

ステップを含む符号化方法。

【請求項 1 4】

係数データを符号化する符号化装置において、

前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数とし、前記最大有効桁数を示す符号として、最大有効桁数が変化したか否かを示す符号を生成する最大有効桁数生成手段と、

前記係数データにより表わされる複数の数値の絶対値を示す符号を生成する絶対値生成手段と、

前記係数データにより表わされる複数の数値の正負を示す符号を生成する符号生成手段と

を備える符号化装置。

【請求項 1 5】

画像データに対して直交変換を行って、前記係数データを生成する直交変換手段と、

前記直交変換手段により生成された係数データを記憶する記憶手段と、

前記記憶手段により記憶された前記係数データを、前記係数データに対して逆直交変換を行う順に予め並び替える係数並び替え手段と

をさらに備える請求項 1 4 に記載の符号化装置。

【請求項 1 6】

画像データに対して、少なくとも最低域成分のサブバンドの 1 ライン分の係数データを生成するのに必要なライン数分の画像データを含むラインブロック毎にフィルタ処理を行い、周波数帯域毎に分解された前記係数データからなる複数のサブバンドを生成するフィルタ手段と、

前記フィルタ手段により生成された係数データをラインブロック毎に記憶する記憶手段と、

前記記憶手段により記憶された前記係数データを、ラインブロック毎に、周波数帯域に分解された複数のサブバンドの係数データを合成して画像データを生成する合成処理を実行する順に予め並び替える係数並び替え手段と

をさらに備える請求項 1 4 に記載の符号化装置。

【請求項 1 7】

前記係数並び替え手段は、前記係数データを、ラインブロック毎に、低域成分から高域成分の順に並び替える

請求項 1 6 に記載の符号化装置。

【請求項 1 8】

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数が増加したかまたは減少したかを示す符号を生成する

請求項 1 4 に記載の符号化装置。

【請求項 1 9】

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数の変化量を示す符号を生成する

10

20

30

40

50

請求項 1 4 に記載の符号化装置。

【請求項 2 0】

前記最大有効桁数生成手段は、連続して並べられた前記係数データの前記最大有効桁数を示す符号を生成し、前記係数データにさらに連続して並べられた前記係数データの最大有効桁数を示す符号を生成する処理を繰り返し行う

請求項 1 4 乃至請求項 1 6 のいずれかに記載の符号化装置。

【請求項 2 1】

前記絶対値生成手段は、前記係数データにより表わされる複数の数値の絶対値を示す符号として、前記係数データが表わす数値の最下位の桁から前記最大有効桁数までの値を示す符号を生成する

請求項 1 4 乃至請求項 1 6 のいずれかに記載の符号化装置。

【請求項 2 2】

前記絶対値生成手段は、前記係数データにより表わされる複数の数値の絶対値を示す符号を並列に生成する

請求項 1 4 乃至請求項 1 6 のいずれかに記載の符号化装置。

【請求項 2 3】

前記係数データにより表される複数の数値が全て 0 であるか否かを示す符号を生成するゼロ符号生成手段をさらに備え、

前記最大有効桁数生成手段は、前記ゼロ符号生成手段により生成される 0 であるか否かを示す符号に示される値に従って、前記最大有効桁数を示す符号を生成する

請求項 1 4 乃至請求項 1 6 のいずれかに記載の符号化装置。

【請求項 2 4】

前記絶対値生成手段は、前記最大有効桁数生成手段により生成される数値の絶対値を示す符号に示される値に従って、数値の絶対値を示す符号を生成し、

前記符号生成手段は、前記絶対値生成手段により生成される数値の絶対値を示す符号に示される値に従って、数値の正負を示す符号を生成する

請求項 2 3 に記載の符号化装置。

【請求項 2 5】

係数データを符号化する符号化方法において、

前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数とし、前記最大有効桁数を示す符号として、最大有効桁数が変化したか否かを示す符号を生成し、

前記係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号を生成し、前記係数データにより表わされるそれぞれの数値の正負を示す複数の符号を生成するステップを含む符号化方法。

【請求項 2 6】

係数データを符号化する符号化装置において、

前記係数データにより表される複数の数値が全て 0 であるか否かを示す符号を生成するゼロ符号生成手段と、

前記ゼロ符号生成手段により生成される 0 であるか否かを示す符号に示される値に従って、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号を生成する最大有効桁数生成手段と、

前記係数データにより表わされる複数の数値の絶対値を示す符号を生成する絶対値生成手段と、

前記係数データにより表わされる複数の数値の正負を示す符号を生成する符号生成手段と

を備える符号化装置。

【請求項 2 7】

画像データに対して直交変換を行って、前記係数データを生成する直交変換手段と、

10

20

30

40

50

前記直交変換手段により生成された係数データを記憶する記憶手段と、  
前記記憶手段により記憶された前記係数データを、前記係数データに対して逆直交変換  
を行う順に予め並び替える係数並び替え手段と  
をさらに備える請求項 2 6 に記載の符号化装置。

【請求項 2 8】

画像データに対して、少なくとも最低域成分のサブバンドの 1 ライン分の係数データを  
生成するのに必要なライン数分の画像データを含むラインブロック毎にフィルタ処理を行  
い、周波数帯域毎に分解された前記係数データからなる複数のサブバンドを生成するフィ  
ルタ手段と、

前記フィルタ手段により生成された係数データをラインブロック毎に記憶する記憶手段  
と、

前記記憶手段により記憶された前記係数データを、ラインブロック毎に、周波数帯域に  
分解された複数のサブバンドの係数データを合成して画像データを生成する合成処理を実  
行する順に予め並び替える係数並び替え手段と

をさらに備える請求項 2 6 に記載の符号化装置。

【請求項 2 9】

前記係数並び替え手段は、前記係数データを、ラインブロック毎に、低域成分から高域  
成分の順に並び替える

請求項 2 8 に記載の符号化装置。

【請求項 3 0】

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す  
符号として、最大有効桁数が増加したかまたは減少したかを示す符号を生成する

請求項 2 6 乃至請求項 2 8 のいずれかに記載の符号化装置。

【請求項 3 1】

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す  
符号として、最大有効桁数の変化量を示す符号を生成する

請求項 3 0 に記載の符号化装置。

【請求項 3 2】

前記絶対値生成手段は、前記最大有効桁数生成手段により生成される数値の絶対値を示  
す符号に示される値に従って、数値の絶対値を示す符号を生成し、

前記符号生成手段は、前記絶対値生成手段により生成される数値の絶対値を示す符号に  
示される値に従って、数値の正負を示す符号を生成する

請求項 2 6 に記載の符号化装置。

【請求項 3 3】

係数データを符号化する符号化方法において、

前記係数データにより表される複数の数値が全て 0 であるか否かを示す符号を生成し、  
生成される 0 であるか否かを示す符号に示される値に従って、前記係数データにより表  
わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大  
有効桁数として、前記最大有効桁数を示す符号を生成し、

前記係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号を生成し、  
前記係数データにより表わされるそれぞれの数値の正負を示す複数の符号を生成する  
ステップを含む符号化方法。

【請求項 3 4】

符号化データを復号して係数データを生成する復号装置において、

前記符号化データを、

前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効  
桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号、

前記係数データにより表わされるそれぞれの数値の絶対値を示す符号、

並びに、前記係数データにより表わされる数値の正負を示す符号  
に分離する符号分離手段と、

10

20

30

40

50

前記符号分離手段により分離されて得られた前記最大有効桁数を示す符号を復号して、前記最大有効桁数を算出する最大有効桁数算出手段と、

前記符号分離手段により分離されて得られた前記絶対値を示す符号を復号して、前記係数データにより表わされる数値の絶対値を算出する絶対値算出手段と、

前記符号分離手段により分離されて得られた前記正負を示す符号を復号して、前記係数データにより表わされるそれぞれの数値の正負を示す符号を生成する符号生成手段と、

前記最大有効桁数算出手段により算出された最大有効桁数、前記絶対値算出手段により算出された絶対値、並びに、前記符号生成手段により生成された正負を示す符号に従って、符号化データを復号して係数データを生成する係数データ生成手段と

を備える復号装置。

10

【請求項 3 5】

前記符号化データを復号して、直交変換することにより得られる前記係数データを生成する復号手段と、

前記復号手段により生成された係数データを記憶する記憶手段と、

前記記憶手段により記憶されている前記係数データを、前記係数データに対して逆直交変換を行う順に並び替える係数並び替え手段と

をさらに備える請求項 3 4 に記載の復号装置。

【請求項 3 6】

前記符号化データを復号して、複数のサブバンドに含まれる前記係数データを生成する復号手段と、

20

前記復号手段により生成された係数データを、少なくとも最低域成分のサブバンドの 1 ライン分の係数データを生成するのに必要なライン数分の画像データを含むラインブロック毎に記憶する記憶手段と、

前記記憶手段により記憶されている前記係数データを、ラインブロック毎に、周波数帯域に分解された複数のサブバンドの係数データを合成する順に並び替える係数並び替え手段と

をさらに備える請求項 3 4 に記載の復号装置。

【請求項 3 7】

前記最大有効桁数算出手段は、最大有効桁数が変化しない場合、前記最大有効桁数を示す符号として、変化がないことを示す符号を復号し、

30

前記最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数が増加したかまたは減少したかを示す符号、および最大有効桁数の変化量を示す符号を復号し、その復号の結果に基づいて前記最大有効桁数を算出する

請求項 3 4 乃至請求項 3 6 のいずれかに記載の復号装置。

【請求項 3 8】

符号化データを復号して係数データを生成する復号方法において、

前記符号化データを、

前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号、

前記係数データにより表わされるそれぞれの数値の絶対値を示す符号、

40

並びに、前記係数データにより表わされる数値の正負を示す符号

に分離し、

分離されて得られる前記最大有効桁数を示す符号を復号して、前記最大有効桁数を算出し、

分離されて得られる前記絶対値を示す符号を復号して、前記係数データにより表わされるそれぞれの数値の絶対値を算出し、

分離されて得られる前記正負を示す符号を復号して、前記係数データにより表わされるそれぞれの数値の正負を示すデータを生成し、算出された最大有効桁数と算出された絶対値と生成された正負を示すデータとに従って、符号化データを復号して係数データを生成する

50

ステップを含む復号方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、符号化装置および方法、並びに、復号装置および方法に関し、特に、より高速に符号化または復号できるようにした符号化装置および方法、並びに、復号装置および方法に関する。

【背景技術】

【0002】

画像（データ）の符号化を行う場合に、入力された画像に対して帯域分割処理を行うことにより生成された各サブバンド（周波数帯域）の係数を符号化する符号化方式として、JPEG（Joint Photographic Experts Group）2000が知られている。

【0003】

JPEG2000方式により画像を符号化する場合、入力された画像をウェーブレット変換することにより得られたウェーブレット係数が量子化され、量子化により得られた量子化係数がさらにエントロピ符号化される。

【0004】

従来、エントロピ符号化においては、EBCOT（Embedded Block Coding with Optimized Truncation）と称されるビットモデリングと、MQコードと称される算術符号化が行われている。すなわち、量子化係数がビットモデリングされ、さらにビットプレーンごとに複数のコーディングパスに基づいて算術符号化される。そして、算術符号化により得られた符号が符号化された画像（データ）として出力される（例えば、特許文献1参照）。

【0005】

また、JPEG2000方式により符号化された画像を復号する場合、符号化を行う場合とは逆の手順で処理が行われる。より具体的には、符号化された画像のデータとしての符号がエントロピ復号されて逆量子化され、これにより得られた量子化係数がさらにウェーブレット逆変換される。そして、ウェーブレット逆変換により得られた画像が復号された画像として出力される。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2004-166254号公報

【発明の概要】

【発明が解決しようとする課題】

【0007】

しかしながら上述した技術では、EBCOTおよびMQコードにおける処理量が多いため、画像の符号化および復号を高速に行うことは困難であり、例えば、横1920画素、縦1080画素などの解像度の高いHD（High Definition）画像をリアルタイムで符号化（または復号）するためには、高価な専用のハードウェアを用意する必要があった。

【0008】

本発明は、このような状況に鑑みてなされたものであり、画像をより高速に符号化または復号することができるようにするものである。

【課題を解決するための手段】

【0009】

本発明の第1の側面の符号化装置は、係数データを符号化する符号化装置において、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号を生成する最大有効桁数生成手段と、前記係数データにより表わされる複数の数値の絶対値を示す符号を生成する絶対値生成手段と、前記係数データにより表わされる複数の数値の正負を示す符号を生成する符号生成手段と、前記最大有効桁数生成手段により生成された最大有効桁数を示

10

20

30

40

50

す符号と、前記絶対値生成手段により生成された絶対値を示す符号と、前記符号生成手段により生成された正負を示す符号とを連結して、符号化データを生成する符号連結手段とを備える。

【0010】

画像データに対して直交変換を行って、前記係数データを生成する直交変換手段と、前記直交変換手段により生成された係数データを記憶する記憶手段と、前記記憶手段により記憶された前記係数データを、前記係数データに対して逆直交変換を行う順に予め並び替える係数並び替え手段とをさらに備えることができる。

【0011】

画像データに対して、少なくとも最低域成分のサブバンドの1ライン分の係数データを生成するのに必要なライン数分の画像データを含むラインブロック毎にフィルタ処理を行い、周波数帯域毎に分解された前記係数データからなる複数のサブバンドを生成するフィルタ手段と、前記フィルタ手段により生成された係数データをラインブロック毎に記憶する記憶手段と、前記記憶手段により記憶された前記係数データを、ラインブロック毎に、周波数帯域に分解された複数のサブバンドの係数データを合成して画像データを生成する合成処理を実行する順に予め並び替える係数並び替え手段とをさらに備えることができる。

10

【0012】

前記係数並び替え手段は、前記係数データを、ラインブロック毎に、低域成分から高域成分の順に並び替えることができる。

20

【0013】

前記最大有効桁数生成手段は、前記最大有効桁数を示す符号として、最大有効桁数が変化したか否かを示す符号を生成することができる。

【0014】

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数が増加したかまたは減少したかを示す符号を生成することができる。

【0015】

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数の変化量を示す符号を生成することができる。

30

【0016】

前記最大有効桁数生成手段は、連続して並べられた前記係数データの前記最大有効桁数を示す符号を生成し、前記係数データにさらに連続して並べられた前記係数データの最大有効桁数を示す符号を生成する処理を繰り返し行うことができる。

【0017】

前記絶対値生成手段は、前記係数データにより表わされる複数の数値の絶対値を示す符号として、前記係数データが表わす数値の最下位の桁から前記最大有効桁数までの値を示す符号を生成することができる。

【0018】

前記絶対値生成手段は、前記係数データにより表わされる複数の数値の絶対値を示す符号を並列に生成することができる。

40

【0019】

前記係数データにより表される複数の数値が全て0であるか否かを示す符号を生成するゼロ符号生成手段をさらに備え、前記最大有効桁数生成手段は、前記ゼロ符号生成手段により生成される0であるか否かを示す符号に示される値に従って、前記最大有効桁数を示す符号を生成することができる。

【0020】

前記絶対値生成手段は、前記最大有効桁数生成手段により生成される数値の絶対値を示す符号に示される値に従って、数値の絶対値を示す符号を生成し、前記符号生成手段は、前記絶対値生成手段により生成される数値の絶対値を示す符号に示される値に従って、数

50



値の正負を示す符号を生成することができる。

【 0 0 2 1 】

本発明の第 1 の側面の符号化方法は、係数データを符号化する符号化方法において、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号を生成し、前記係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号を生成し、前記係数データにより表わされるそれぞれの数値の正負を示す複数の符号を生成し、生成された最大有効桁数を示す符号と、生成された絶対値を示す符号と、生成された正負を示す符号とを連結して、符号化データを生成するステップを含む。

【 0 0 2 2 】

本発明の第 2 の側面の符号化装置は、係数データを符号化する符号化装置において、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数とし、前記最大有効桁数を示す符号として、最大有効桁数が変化したか否かを示す符号を生成する最大有効桁数生成手段と、前記係数データにより表わされる複数の数値の絶対値を示す符号を生成する絶対値生成手段と、前記係数データにより表わされる複数の数値の正負を示す符号を生成する符号生成手段とを備える。

画像データに対して直交変換を行って、前記係数データを生成する直交変換手段と、前記直交変換手段により生成された係数データを記憶する記憶手段と、前記記憶手段により記憶された前記係数データを、前記係数データに対して逆直交変換を行う順に予め並び替える係数並び替え手段とをさらに備えることができる。

画像データに対して、少なくとも最低域成分のサブバンドの 1 ライン分の係数データを生成するのに必要なライン数分の画像データを含むラインブロック毎にフィルタ処理を行い、周波数帯域毎に分解された前記係数データからなる複数のサブバンドを生成するフィルタ手段と、前記フィルタ手段により生成された係数データをラインブロック毎に記憶する記憶手段と、前記記憶手段により記憶された前記係数データを、ラインブロック毎に、周波数帯域に分解された複数のサブバンドの係数データを合成して画像データを生成する合成処理を実行する順に予め並び替える係数並び替え手段とをさらに備えることができる。

前記係数並び替え手段は、前記係数データを、ラインブロック毎に、低域成分から高域成分の順に並び替えることができる。

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数が増加したかまたは減少したかを示す符号を生成することができる。

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数の変化量を示す符号を生成することができる。

前記最大有効桁数生成手段は、連続して並べられた前記係数データの前記最大有効桁数を示す符号を生成し、前記係数データにさらに連続して並べられた前記係数データの最大有効桁数を示す符号を生成する処理を繰り返し行うことができる。

前記絶対値生成手段は、前記係数データにより表わされる複数の数値の絶対値を示す符号として、前記係数データが表わす数値の最下位の桁から前記最大有効桁までの値を示す符号を生成することができる。

前記絶対値生成手段は、前記係数データにより表わされる複数の数値の絶対値を示す符号を並列に生成することができる。

前記係数データにより表わされる複数の数値が全て 0 であるか否かを示す符号を生成するゼロ符号生成手段をさらに備え、前記最大有効桁数生成手段は、前記ゼロ符号生成手段により生成される 0 であるか否かを示す符号に示される値に従って、前記最大有効桁数を示す符号を生成することができる。

前記絶対値生成手段は、前記最大有効桁数生成手段により生成される数値の絶対値を示す符号に示される値に従って、数値の絶対値を示す符号を生成し、前記符号生成手段は、前記絶対値生成手段により生成される数値の絶対値を示す符号に示される値に従って、数

10

20

30

40

50

値の正負を示す符号を生成することができる。

本発明の第2の側面の符号化方法は、係数データを符号化する符号化方法において、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数とし、前記最大有効桁数を示す符号として、最大有効桁数が変化したか否かを示す符号を生成し、前記係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号を生成し、前記係数データにより表わされるそれぞれの数値の正負を示す複数の符号を生成するステップを含む。

本発明の第3の側面の符号化装置は、係数データを符号化する符号化装置において、前記係数データにより表される複数の数値が全て0であるか否かを示す符号を生成するゼロ符号生成手段と、前記ゼロ符号生成手段により生成される0であるか否かを示す符号に示される値に従って、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号を生成する最大有効桁数生成手段と、前記係数データにより表わされる複数の数値の絶対値を示す符号を生成する絶対値生成手段と、前記係数データにより表わされる複数の数値の正負を示す符号を生成する符号生成手段とを備える。

画像データに対して直交変換を行って、前記係数データを生成する直交変換手段と、前記直交変換手段により生成された係数データを記憶する記憶手段と、前記記憶手段により記憶された前記係数データを、前記係数データに対して逆直交変換を行う順に予め並び替える係数並び替え手段とをさらに備えることができる。

画像データに対して、少なくとも最低域成分のサブバンドの1ライン分の係数データを生成するのに必要なライン数分の画像データを含むラインブロック毎にフィルタ処理を行い、周波数帯域毎に分解された前記係数データからなる複数のサブバンドを生成するフィルタ手段と、前記フィルタ手段により生成された係数データをラインブロック毎に記憶する記憶手段と、前記記憶手段により記憶された前記係数データを、ラインブロック毎に、周波数帯域に分解された複数のサブバンドの係数データを合成して画像データを生成する合成処理を実行する順に予め並び替える係数並び替え手段とをさらに備えることができる。

前記係数並び替え手段は、前記係数データを、ラインブロック毎に、低域成分から高域成分の順に並び替えることができる。

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数が増加したかまたは減少したかを示す符号を生成することができる。

前記最大有効桁数生成手段は、最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数の変化量を示す符号を生成することができる。

前記絶対値生成手段は、前記最大有効桁数生成手段により生成される数値の絶対値を示す符号に示される値に従って、数値の絶対値を示す符号を生成し、前記符号生成手段は、前記絶対値生成手段により生成される数値の絶対値を示す符号に示される値に従って、数値の正負を示す符号を生成することができる。

本発明の第3の側面の符号化方法は、係数データを符号化する符号化方法において、前記係数データにより表される複数の数値が全て0であるか否かを示す符号を生成し、生成される0であるか否かを示す符号に示される値に従って、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号を生成し、前記係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号を生成し、前記係数データにより表わされるそれぞれの数値の正負を示す複数の符号を生成するステップを含む。

【0023】

本発明の第4の側面の復号装置は、符号化データを復号して係数データを生成する復号装置において、前記符号化データを、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号、前記係数データにより表わされるそれぞれの数値の絶対値を示す符

号、並びに、前記係数データにより表わされる数値の正負を示す符号に分離する符号分離手段と、前記符号分離手段により分離されて得られた前記最大有効桁数を示す符号を復号して、前記最大有効桁数を算出する最大有効桁数算出手段と、前記符号分離手段により分離されて得られた前記絶対値を示す符号を復号して、前記係数データにより表わされる数値の絶対値を算出する絶対値算出手段と、前記符号分離手段により分離されて得られた前記正負を示す符号を復号して、前記係数データにより表わされるそれぞれの数値の正負を示す符号を生成する符号生成手段と、前記最大有効桁数算出手段により算出された最大有効桁数と前記絶対値算出手段により算出された絶対値と前記符号生成手段により生成された正負を示す符号とに従って、符号化データを復号して係数データを生成する係数データ生成手段とを備える。

10

#### 【0024】

前記符号化データを復号して、直交変換することにより得られる前記係数データを生成する復号手段と、前記復号手段により生成された係数データを記憶する記憶手段と、前記記憶手段により記憶されている前記係数データを、前記係数データに対して逆直交変換を行う順に並び替える係数並び替え手段とをさらに備えることができる。

#### 【0025】

前記符号化データを復号して、複数のサブバンドに含まれる前記係数データを生成する復号手段と、前記復号手段により生成された係数データを、少なくとも最低域成分のサブバンドの1ライン分の係数データを生成するのに必要なライン数分の画像データを含むラインブロック毎に記憶する記憶手段と、前記記憶手段により記憶されている前記係数データを、ラインブロック毎に、周波数帯域に分解された複数のサブバンドの係数データを合成する順に並び替える係数並び替え手段とをさらに備えることができる。

20

#### 【0027】

前記最大有効桁数算出手段は、最大有効桁数が変化しない場合、前記最大有効桁数を示す符号として、変化がないことを示す符号を復号し、前記最大有効桁数が変化した場合、前記最大有効桁数を示す符号として、最大有効桁数が増加したかまたは減少したかを示す符号、および最大有効桁数の変化量を示す符号を復号し、その復号の結果に基づいて前記最大有効桁数を算出することができる。

#### 【0028】

本発明の第4の側面の復号方法は、前記符号化データを、前記係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を前記係数データの最大有効桁数として、前記最大有効桁数を示す符号、前記係数データにより表わされるそれぞれの数値の絶対値を示す符号、並びに、前記係数データにより表わされる数値の正負を示す符号に分離し、分離されて得られる前記最大有効桁数を示す符号を復号して、前記最大有効桁数を算出し、分離されて得られる前記絶対値を示す符号を復号して、前記係数データにより表わされるそれぞれの数値の絶対値を算出し、分離されて得られる前記正負を示す符号を復号して、前記係数データにより表わされるそれぞれの数値の正負を示すデータを生成し、算出された最大有効桁数と算出された絶対値と生成された正負を示すデータとに従って、符号化データを復号して係数データを生成するステップを含む。

30

#### 【0029】

本発明の第1の側面においては、係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を係数データの最大有効桁数として、最大有効桁数を示す符号が生成され、係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号が生成され、係数データにより表わされるそれぞれの数値の正負を示す複数の符号が生成され、生成された最大有効桁数を示す符号と、生成された絶対値を示す符号と、生成された正負を示す符号とが連結されて、符号化データが生成される。

40

本発明の第2の側面においては、係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を係数データの最大有効桁数とし、最大有効桁数を示す符号として、最大有効桁数が変化したか否かを示す符号が生成され、係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号が生成され、係数データにより表わされ

50

るそれぞれの数値の正負を示す複数の符号が生成される。

本発明の第3の側面においては、係数データにより表される複数の数値が全て0であるか否かを示す符号が生成され、生成される0であるか否かを示す符号に示される値に従って、係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を係数データの最大有効桁数として、最大有効桁数を示す符号が生成され、係数データにより表わされるそれぞれの数値の絶対値を示す複数の符号が生成され、係数データにより表わされるそれぞれの数値の正負を示す複数の符号が生成される。

【0030】

本発明の第4の側面においては、符号化データが、係数データにより表わされる複数の数値のうち、絶対値が最も大きい数値の有効桁数を係数データの最大有効桁数として、最大有効桁数を示す符号、係数データにより表わされるそれぞれの数値の絶対値を示す符号、並びに、係数データにより表わされる数値の正負を示す符号に分離され、分離されて得られる最大有効桁数を示す符号が復号されて、最大有効桁数が算出され、分離されて得られる絶対値を示す符号が復号されて、係数データにより表わされるそれぞれの数値の絶対値が算出され、分離されて得られる正負を示す符号が復号されて、係数データにより表わされるそれぞれの数値の正負を示すデータが生成され、算出された最大有効桁数と算出された絶対値と生成された正負を示すデータとに従って、符号化データが復号されて係数データが生成される。

【発明の効果】

【0031】

本発明によれば、画像を符号化することができる。特に、より高速に画像を符号化することができる。

【0032】

また、本発明によれば、画像を復号することができる。特に、より高速に画像を復号することができる。

【図面の簡単な説明】

【0033】

【図1】本発明を適用した画像符号化装置の構成例を示すブロック図である。

【図2】サブバンドを説明するための図である。

【図3】符号化される量子化係数の一例を示す図である。

【図4】エントロピ符号化部の構成例を示すブロック図である。

【図5】符号化処理を説明するためのフローチャートである。

【図6】エントロピ符号化処理を説明するためのフローチャートである。

【図7】w個組符号化処理を説明するためのフローチャートである。

【図8】画像復号装置の構成例を示すブロック図である。

【図9】エントロピ復号部の構成例を示すブロック図である。

【図10】符号分割部の構成例を示すブロック図である。

【図11】符号分割部の他の構成例を示すブロック図である。

【図12】復号処理を説明するためのフローチャートである。

【図13】エントロピ復号処理を説明するためのフローチャートである。

【図14】w個組復号処理を説明するためのフローチャートである。

【図15】エントロピ符号化部の他の構成例を示すブロック図である。

【図16】符号化される量子化係数の一例を示す図である。

【図17】w個組符号化処理を説明するためのフローチャートである。

【図18】w個組復号処理を説明するためのフローチャートである。

【図19】本発明を適用した画像符号化装置の一例の構成を示すブロック図である。

【図20】ウェーブレット変換について概略的に説明するための略線図である。

【図21】ウェーブレット変換について概略的に説明するための略線図である。

【図22】5×3フィルタに対してリフティング技術を適用した場合のウェーブレット変換について概略的に説明するための略線図である。

10

20

30

40

50

【図23】 $5 \times 3$ フィルタに対してリフティング技術を適用した場合のウェーブレット逆変換について概略的に説明するための略線図である。

【図24】 $5 \times 3$ フィルタのリフティングによるフィルタリングを分解レベル=2まで実行した例を示す略線図である。

【図25】この発明によるウェーブレット変換およびウェーブレット逆変換の流れを概略的に示す略線図である。

【図26】符号化処理の流れの例を説明するためのフローチャートである。

【図27】本発明を適用した画像復号装置の一例の構成を示すブロック図である。

【図28】復号処理の流れの例を説明するためのフローチャートである。

【図29】本発明を適用した画像符号化装置および画像復号装置の各要素の一例の並列動作を概略的に示す略線図である。 10

【図30】本発明を適用した画像符号化装置の一例の構成を示すブロック図である。

【図31】ウェーブレット係数の並び替え処理を画像符号化装置側で行う場合の処理の流れを説明するための略線図である。

【図32】ウェーブレット係数の並び替え処理を画像復号装置側で行う場合の処理の流れを説明するための略線図である。

【図33】本発明を適用した画像符号化装置の一例の構成を示すブロック図である。

【図34】本発明を適用した画像復号装置の一例の構成を示すブロック図である。

【図35】符号化データの授受の様子を説明する模式図である。

【図36】パケットの構成例を示す図である。 20

【図37】本発明を適用したデジタルライアックスシステムの一例の構成を示すブロック図である。

【図38】本発明を適用した無線伝送システムの一例の構成を示すブロック図である。

【図39】本発明を適用した家庭用ゲーム機器の一例の構成を示す図である。

【図40】本発明を適用した情報処理システムの構成例を示す図である。

【発明を実施するための形態】

【0034】

図1は、本発明を適用した画像符号化装置の構成例を示すブロック図である。

【0035】

画像符号化装置11は、ウェーブレット変換部21、量子化部22、およびエントロピ符号化部23を有する。 30

【0036】

ウェーブレット変換部21には、例えば、必要に応じてDCレベルシフトが施されたコンポーネント信号である画像（データ）が入力される。ウェーブレット変換部21は、その入力された画像をウェーブレット変換して複数のサブバンドに分解する。ウェーブレット変換部21は、ウェーブレット変換により得られたサブバンドのウェーブレット係数を量子化部22に供給する。

【0037】

量子化部22は、ウェーブレット変換部21から供給されたウェーブレット係数を量子化し、その結果として得られた量子化係数をエントロピ符号化部23に供給する。 40

【0038】

エントロピ符号化部23は、量子化部22から供給された量子化係数をエントロピ符号化し、これにより得られた符号を符号化された画像（データ）として出力する。エントロピ符号化部23から出力された画像は、例えば、レート制御処理された後、パケット化されて記録されたり、画像符号化装置11に接続された他の装置（図示せず）に供給されたりする。

【0039】

次に、図2および図3を参照して、図1のエントロピ符号化部23が行うエントロピ符号化について説明する。

【0040】

例えば、図2に示すように、1つのサブバンドがラインL1乃至ラインL6の6本のラインか 50

ら構成されているとし、xy座標系におけるライン上の画素に対応する位置を $(x, y)$ とする。ここで、各ラインの図中、左端の位置のx座標は0とされ、ラインL1のy座標は0とされる。

【0041】

量子化部22からエントロピ符号化部23には、ビットプレーン表現された、サブバンドの各位置 $(x, y)$ における量子化係数がラインL1からラインL6までラスタスキャン順に入力される。

【0042】

換言すれば、エントロピ符号化部23には、まず、ラインL1の左端の位置 $(0, 0)$ に対応する量子化係数が入力される。次にその位置 $(0, 0)$ の右隣の位置 $(1, 0)$ に対応する量子化係数がエントロピ符号化部23に入力され、ラインL1の右端の位置まで、量子化係数が入力された位置の右隣の位置に対応する量子化係数がエントロピ符号化部23に順次入力される。そして、ラインL1上の位置の量子化係数が全て入力されると、ラインL2の左端の位置 $(0, 1)$ から順番に右端の位置まで、ラインL2上の各位置に対応する量子化係数がエントロピ符号化部23に入力され、同様にラインL3からラインL6まで、各ライン上の位置に対応する量子化係数がエントロピ符号化部23に入力される。

10

【0043】

例えば、図3の図中、左上に示すように、図2のラインL1の左端の位置の量子化係数から順番に、12個の量子化係数がエントロピ符号化部23に入力されると、エントロピ符号化部23は、予め定められた所定の数 $w$ （図3では $w=4$ ）ずつ量子化係数を符号化する。

20

【0044】

ここで、図3の左上に示された各量子化係数は、その符号の絶対値が2進数の桁に分けられて表現されて（ビットプレーン表現されて）おり、図3の例では、エントロピ符号化部23には、1つのライン（図2のラインL1）の量子化係数“-0101”、“+0011”、“-0110”、“+0010”、“+0011”、“+0110”、“0000”、“-0011”、“+1101”、“-0100”、“+0111”、および“-1010”が順番に入力される。

【0045】

1つの量子化係数は、“+”（正）または“-”（負）で表わされる量子化係数の符号（以下、量子化係数のサイン（Sign）と称する。）と、2進数で表わされた量子化係数の絶対値とからなる。図3では、量子化係数の絶対値の各桁の値を示す各ビットのうち、図中、最も上側のビットが最上位ビット（最上位の桁のビット）を表わしている。したがって、例えば、量子化係数“-0101”は、そのサインが“-”であり、2進数で表わされた絶対値が“0101”であるので、この量子化係数は10進数で表わすと“-5”となる。

30

【0046】

まず、エントロピ符号化部23は、入力された1つのラインの量子化係数（の絶対値）が全て0であるか否かを判定し、その判定結果に応じて、これから符号化するラインの量子化係数が全て0であるか否かを示す符号を出力する。量子化係数が全て0であると判定した場合、エントロピ符号化部23は、ラインの量子化係数が全て0であるか否かを示す符号として0を出力して、現在行っているラインの量子化係数の符号化を終了する。また、全ての量子化係数の値が0ではない（0の量子化係数のみではない）と判定した場合、エントロピ符号化部23は、ラインの量子化係数が全て0であるか否かを示す符号として1を出力する。

40

【0047】

図中、左上に示した12個の量子化係数が入力された場合、入力されたラインの量子化係数は0のみではないので、図中、右上に示すように、エントロピ符号化部23は符号として1を出力する。

【0048】

ラインの量子化係数が全て0であるか否かを示す符号として、量子化係数が全て0でないことを示す符号1が出力されると、次に、エントロピ符号化部23は、入力された最初の4つ（ $w$ 個）の量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の符号化を行

50

う。

【 0 0 4 9 】

エントロピ符号化部23は、今回入力された連続する4つの量子化係数の最大有効桁数（図3における変数Bの値）と、前回符号化した（入力された）4つ（w個）の量子化係数の最大有効桁数とを比較し、最大有効桁数が変化したか否かを判定して、量子化係数の最大有効桁数を示す符号を出力する。

【 0 0 5 0 】

ここで、最大有効桁数とは、まとめて符号化する4つ（w個）の量子化係数のうち、絶対値が最も大きい量子化係数の有効桁数をいう。換言すれば、最大有効桁数は、4つの量子化係数のうち、絶対値が最も大きい量子化係数の最上位にある1が何桁目にあるかを示す。したがって、例えば、まとめて符号化する4つの量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の最大有効桁数は、絶対値が最も大きい量子化係数“-0110”の最上位にある1の桁である“3”とされる。

【 0 0 5 1 】

また、量子化係数の最大有効桁数を示す符号は、最大有効桁数が変化したか否かを示す符号、最大有効桁数が増加したか、または減少したかを示す符号、および最大有効桁数の変化量を示す符号からなり、最大有効桁数が変化していない場合、最大有効桁数が増加したか、または減少したかを示す符号、および最大有効桁数の変化量を示す符号は出力されない。

【 0 0 5 2 】

エントロピ符号化部23は、最大有効桁数の比較の結果、最大有効桁数が変化した場合、最大有効桁数が変化したことを示す符号1を出力し、最大有効桁数が変化していない場合、最大有効桁数が変化していないことを示す符号0を出力する。

【 0 0 5 3 】

なお、最大有効桁数が変化したか否かを判定する場合に、今回初めて4つの量子化係数が入力されるとき、すなわち、符号化するサブバンドの量子化係数が初めて入力されるとき（例えば、図2のラインL1の左端から順番に4つの量子化係数が入力されるとき）、前回、そのサブバンドの量子化係数は符号化されていないので、前回符号化した4つ（w個）の量子化係数の最大有効桁数は0とされる。

【 0 0 5 4 】

したがって、エントロピ符号化部23は、今回入力された4つの量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の最大有効桁数3と、前回符号化した量子化係数の最大有効桁数0とを比較し、最大有効桁数が変化したので符号1を出力する。

【 0 0 5 5 】

また、エントロピ符号化部23は、最大有効桁数が変化したことを示す符号1に続いて、最大有効桁数が増加したか、または減少したかを示す符号を出力する。ここで、エントロピ符号化部23は、最大有効桁数が増加した場合には0を出力し、最大有効桁数が減少した場合には1を出力する。

【 0 0 5 6 】

前回の最大有効桁数は0であり、今回の最大有効桁数は3であるので、図中、右上に示すように、エントロピ符号化部23は、最大有効桁数が増加したことを示す符号0を出力する。

【 0 0 5 7 】

さらに、エントロピ符号化部23は、最大有効桁数が増加したか、または減少したかを示す符号を出力すると、最大有効桁数がどれだけ増加または減少したかを示す符号、すなわち、最大有効桁数の変化量を示す符号を出力する。具体的には、エントロピ符号化部23は、最大有効桁数の変化量（すなわち、増加量または減少量）をnとすると、（n-1）個の符号0を出力し、それらの0に続いて符号1を出力する。

【 0 0 5 8 】

図3の最初の4つの量子化係数を符号化する場合、最大有効桁数の変化量は3（=3-0）で

あるので、エントロピ符号化部23は符号として、2 (= 3-1) 個の0を出力し、さらに1を出力する。

【 0 0 5 9 】

次に、エントロピ符号化部23は、今回符号化する4つ (w個) の量子化係数のそれぞれの絶対値を示す最大有効桁数分の符号を出力する。すなわち、エントロピ符号化部23は、それぞれの量子化係数について、最大有効桁数により示される有効桁の最大の桁から順番に最小の桁まで、量子化係数の絶対値の各桁の値を示す符号を出力する。

【 0 0 6 0 】

今回符号化する量子化係数は、“-0101”、“+0011”、“-0110”、および“+0010”があるので、エントロピ符号化部23は、まず、最初に入力された量子化係数“-0101”の絶対値を示す最大有効桁数分の符号を出力する。ここで、今回の最大有効桁数は3であるので、エントロピ符号化部23は、量子化係数“-0101”の最大有効桁数により示される有効桁の最大の桁 (すなわち、3桁目) の値“1”、最大の桁より1つ下の桁 (2桁目) の値“0”、および最下位の桁の値“1”を出力する。これにより、量子化係数“-0101”の絶対値を示す有効桁数分の符号“101”が出力される。

【 0 0 6 1 】

同様に、エントロピ符号化部23は、量子化係数“+0011”、“-0110”、および“+0010”の絶対値を示す有効桁数分の符号“011”、“110”、および“010”を順番に出力する。したがって、量子化係数は、“-0101”、“+0011”、“-0110”、および“+0010”のそれぞれの絶対値を示す最大有効桁数分の符号として、“101011110010”が出力される。このように、エントロピ符号化部23からは、量子化係数の絶対値を示す符号として、符号化する4つの量子化係数の最大有効桁数に応じた長さの符号が出力される。

【 0 0 6 2 】

そして最後に、エントロピ符号化部23は、4つ (w個) の量子化係数のうち、絶対値が0でない量子化係数のそれぞれのサインを示す符号を出力する。ここで、エントロピ符号化部23は、量子化係数のサインが“+” (正) である場合、符号0を出力し、サインが“-” (負) である場合、符号1を出力する。

【 0 0 6 3 】

今回符号化する量子化係数は、“-0101”、“+0011”、“-0110”、および“+0010”であり、これらの量子化係数のサインは、順に負、正、負、正であるので、図中、右上に示すように、エントロピ符号化部23は、量子化係数のそれぞれのサインを示す符号として、“1010”を出力する。

【 0 0 6 4 】

最初に入力された4つの量子化係数が符号化されると、エントロピ符号化部23は、続いて、次の連続する4つの量子化係数“+0011”、“+0110”、“0000”、および“-0011”の符号化を行う。

【 0 0 6 5 】

最初に (前回) 入力された量子化係数の符号化における場合と同様に、まず、エントロピ符号化部23は、今回、新たに入力された4つ (w個) の量子化係数の最大有効桁数と、前回符号化した4つの量子化係数の最大有効桁数とを比較する。

【 0 0 6 6 】

今回入力された4つ (w個) の量子化係数“+0011”、“+0110”、“0000”、および“-0011”の最大有効桁数は、絶対値が最も大きい量子化係数“+0110”の最上位にある1の桁である“3”であり、前回符号化した量子化係数の最大有効桁数“3”と同じであるので、エントロピ符号化部23は、最大有効桁数が変化していないことを示す符号0を出力する。

【 0 0 6 7 】

続いて、エントロピ符号化部23は、今回符号化する4つ (w個) の量子化係数“+0011”、“+0110”、“0000”、および“-0011”のそれぞれの絶対値を示す最大有効桁数分の符号“011”、“110”、“000”、および“011”が順番に並べられた符号“011110000011”を出力する。



## 【 0 0 6 8 】

そして、量子化係数の絶対値を示す符号が出力されると、エントロピ符号化部23は、4つの量子化係数のうち、絶対値が0でない量子化係数のそれぞれのサインを示す符号を出力する。

## 【 0 0 6 9 】

今回符号化する量子化係数は、“+0011”、“+0110”、“0000”、および“-0011”であり、3つ目の量子化係数“0000”はその絶対値が0であるので、エントロピ符号化部23は、0でない量子化係数“+0011”、“+0110”、および“-0011”のそれぞれのサイン（正、正、負）を示す符号“001”を出力する。

## 【 0 0 7 0 】

4つの量子化係数“+0011”、“+0110”、“0000”、および“-0011”が符号化されると、さらに、エントロピ符号化部23は、次の4つの量子化係数“+1101”、“-0100”、“+0111”、および“-1010”の符号化を行う。

## 【 0 0 7 1 】

まず、エントロピ符号化部23は、今回、新たに入力された4つ（w個）の量子化係数の最大有効桁数と、前回符号化した4つの量子化係数の最大有効桁数とを比較する。

## 【 0 0 7 2 】

今回入力された4つ（w個）の量子化係数“+1101”、“-0100”、“+0111”、および“-1010”の最大有効桁数は、絶対値が最も大きい量子化係数“+1101”の最上位にある1の桁である“4”であり、前回符号化した量子化係数の最大有効桁数“3”とは異なるので、エントロピ符号化部23は、最大有効桁数が変化したことを示す符号1を出力する。

## 【 0 0 7 3 】

また、前回の最大有効桁数は3であり、今回の最大有効桁数は4であるので、エントロピ符号化部23は、図中、右側に示すように、最大有効桁数が増加したことを示す符号0を出力する。

## 【 0 0 7 4 】

さらに、エントロピ符号化部23は、最大有効桁数がどれだけ増加または減少したかを示す符号を出力する。この場合、最大有効桁数の変化量は1（=4-3）であるので、エントロピ符号化部23は符号として、0（=1-1）個の0を出力し、さらに1を出力する（すなわち、符号1を出力する）。

## 【 0 0 7 5 】

次に、エントロピ符号化部23は、今回符号化する4つ（w個）の量子化係数“+1101”、“-0100”、“+0111”、および“-1010”のそれぞれの絶対値を示す最大有効桁数分の符号“1101”、“0100”、“0111”、および“1010”が順番に並べられた符号“110101000111010”を出力する。

## 【 0 0 7 6 】

そして、量子化係数の絶対値を示す符号が出力されると、エントロピ符号化部23は、4つの量子化係数のうち、0でない量子化係数のそれぞれのサインを示す符号を出力する。

## 【 0 0 7 7 】

今回符号化する量子化係数は、“+1101”、“-0100”、“+0111”、および“-1010”であり、これらの量子化係数のサインは、順番に正、負、正、負であるので、図中、右下に示すように、エントロピ符号化部23は、量子化係数のそれぞれのサインを示す符号として、“0101”を出力する。

## 【 0 0 7 8 】

このようにして、エントロピ符号化部23は、入力された量子化係数を、連続する予め定められた数（w個）ずつ符号化する。これにより、エントロピ符号化部23からは、符号化するラインの量子化係数が全て0であるか否かを示す符号が出力され、ラインの量子化係数が全て0でないことを示す符号が出力されると、次に、w個の量子化係数の最大有効桁数を示す符号、w個の量子化係数の絶対値（ビットプレーン表現）を示す符号、およびそれらの量子化係数のサインを示す符号が出力される。

## 【 0 0 7 9 】

そして、これらのw個の量子化係数の最大有効桁数を示す符号、w個の量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号のそれぞれは、そのラインの量子化係数が全て符号化されるまで、次のw個の量子化係数の最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号が繰り返し出力される。

## 【 0 0 8 0 】

なお、量子化係数がラスタスキャン順に符号化されると説明したが、量子化係数が符号化される順番は必ずしもラスタスキャン順である必要はない。例えば、図2に示したサブバンドの量子化係数が符号化される場合、最初に位置(0,0)、(0,1)、(0,2)、および(0,3)（すなわち、ラインL1乃至ラインL4のそれぞれの図中、左端の位置）の量子化係数が符号化され、次に、位置(1,0)、(1,1)、(1,2)、および(1,3)の量子化係数が符号化されるといったように、図中、縦方向に並ぶ4つの位置の量子化係数をw個の量子化係数として、w個ずつ順番に符号化するようにしてもよい。

10

## 【 0 0 8 1 】

以上において説明した処理を行う図1のエントロピ符号化部23は、より詳細には、図4に示すように構成される。

## 【 0 0 8 2 】

エントロピ符号化部23は、ライン判定部61、VLC (Variable Length Coding) 符号化部62、最大有効桁数計算部63、VLC符号化部64、有効桁抽出部65、VLC符号化部66、サイン抽出部67、VLC符号化部68、および符号連結部69を有する。

20

## 【 0 0 8 3 】

量子化部22（図1）から出力された量子化係数は、ライン判定部61、最大有効桁数計算部63、有効桁抽出部65、およびサイン抽出部67に供給（入力）される。

## 【 0 0 8 4 】

ライン判定部61は、量子化部22から入力された、これから符号化する1つのラインの量子化係数が全て0であるか否かを判定し、その判定の結果を示す情報をVLC符号化部62に供給する。

## 【 0 0 8 5 】

VLC符号化部62は、ライン判定部61からの判定の結果を示す情報に基づいて、符号化するラインの量子化係数が全て0であるか否かを示す符号を符号連結部69に出力する。

30

## 【 0 0 8 6 】

最大有効桁数計算部63は、量子化部22から入力された、連続するw個の量子化係数の最大有効桁数を計算し、その計算の結果を示す情報をVLC符号化部64および有効桁抽出部65に供給する。

## 【 0 0 8 7 】

VLC符号化部64は、最大有効桁数計算部63からの計算の結果を示す情報に基づいて、w個の量子化係数の最大有効桁数を示す符号を符号連結部69に供給する。

## 【 0 0 8 8 】

有効桁抽出部65は、最大有効桁数計算部63からの計算の結果を示す情報に基づいて、量子化部22から供給されたw個の量子化係数の有効桁を抽出し、抽出した量子化係数の有効桁（のデータ）をVLC符号化部66およびサイン抽出部67に供給する。

40

## 【 0 0 8 9 】

VLC符号化部66は、有効桁抽出部65からの量子化係数の有効桁に基づいて、これらの量子化係数の絶対値を符号化し、これにより得られた量子化係数の絶対値を示す符号を符号連結部69に供給する。

## 【 0 0 9 0 】

サイン抽出部67は、有効桁抽出部65からの量子化係数の有効桁に基づいて、量子化部22から供給された量子化係数のサインを抽出して、抽出したサイン（のデータ）をVLC符号化部68に供給する。

## 【 0 0 9 1 】

50

VLC符号化部68は、サイン抽出部67からのサイン（のデータ）を符号化し、これにより得られる量子化係数のサインを示す符号を符号連結部69に供給する。

【0092】

符号連結部69は、VLC符号化部62、VLC符号化部64、VLC符号化部66、およびVLC符号化部68のそれぞれから供給された、ラインの量子化係数が全て0であるか否かを示す符号、最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号のそれぞれを連結し、符号化された画像（データ）として出力する。

【0093】

次に、図5のフローチャートを参照して、画像符号化装置11（図1）による、符号化処理について説明する。この符号化処理は、ウェーブレット変換部21に、符号化する画像（データ）が入力されると開始される。

10

【0094】

ステップS11において、ウェーブレット変換部21は、入力された画像にウェーブレット変換を施して、入力された画像を複数のサブバンドに分解し、各サブバンドのウェーブレット係数を量子化部22に供給する。

【0095】

ステップS12において、量子化部22は、ウェーブレット変換部21から供給されたウェーブレット係数を量子化し、その結果得られた量子化係数をエントロピ符号化部23に供給する。これにより、エントロピ符号化部23には、例えば、図3を参照して説明したビットプレーン表現された、サブバンドの各位置の量子化係数が入力される。

20

【0096】

ステップS13において、エントロピ符号化部23は、エントロピ符号化処理を行い、符号化処理を終了する。なお、エントロピ符号化処理の詳細は後述するが、エントロピ符号化部23は、エントロピ符号化処理において、図3を参照して説明したように、量子化部22から供給された量子化係数を所定の数（ $w$ 個）ずつ符号化し、符号化するラインの量子化係数が全て0であるか否かを示す符号、量子化係数の最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号を、符号化された画像（データ）として出力する。

【0097】

このようにして、画像符号化装置11は、入力された画像を符号化して出力する。

30

【0098】

次に、図6のフローチャートを参照して、図5のステップS13の処理に対応するエントロピ符号化処理について説明する。

【0099】

図5のステップS12において、量子化部22から出力された量子化係数は、エントロピ符号化部23（図4）のライン判定部61、最大有効桁数計算部63、有効桁抽出部65、およびサイン抽出部67に供給（入力）される。

【0100】

ステップS41において、ライン判定部61は、これから符号化するサブバンドのラインを示す変数 $y$ を $y=0$ として、これを記憶する。

40

【0101】

例えば、図2に示したサブバンドの量子化係数を符号化する場合、ライン判定部61は、そのサブバンドのライン（ラインL1乃至ラインL6）を示す変数 $y$ を、 $y=0$ とする。なお、ここで、変数 $y$ により示されるライン $y$ は、サブバンドのライン上の各位置（ $x, y$ ）の $y$ 座標が $y$ であるラインを示している。したがって、例えば、ライン判定部61が記憶している変数 $y$ が $y=0$ である場合、その変数により示されるラインは、ライン上の各位置の $y$ 座標が0であるラインL1となる。

【0102】

ステップS42において、最大有効桁数計算部63は、ライン判定部61が記憶している変数 $y$ により示されるライン $y$ よりも1つ前のライン（ $y-1$ ）上の最初に入力される $w$ 個の量子化係

50

数の最大有効桁数を示す変数BinitをBinit = 0として、これを記憶する。

【 0 1 0 3 】

例えば、ライン (y - 1) が、図2に示したラインL1である場合、ライン (y - 1) 上の最初に入力されるw個の量子化係数の最大有効桁数を示す変数Binitの値は、ラインL1の図中、左端の位置からw個の量子化係数、すなわち、位置 (0, 0)、(1, 0)、・・・、(w-1, 0) のw個の量子化係数の最大有効桁数となる。また、ライン判定部61が記憶している変数yがy = 0である場合、ライン (y - 1) は存在しないので、変数Binitの値は、Binit = 0とされる。

【 0 1 0 4 】

ステップS43において、ライン判定部61は、記憶している変数yにより示されるラインyの量子化係数 (の絶対値) が全て0であるか否かを判定する。例えば、ラインyが、図2に示したラインL1である場合、ライン判定部61は、ラインL1上の位置 (x, y) の量子化係数が全て0であるとき、量子化係数が全て0であると判定する。

【 0 1 0 5 】

ステップS43において、量子化係数が全て0であると判定した場合、ライン判定部61は、量子化係数が全て0である旨の情報を生成してそれをVLC符号化部62および最大有効桁数計算部63に供給し、処理をステップS44に進める。

【 0 1 0 6 】

ステップS44において、VLC符号化部62は、ライン判定部61からの量子化係数が全て0である旨の情報に基づいて、符号化するラインの量子化係数が全て0であることを示す符号0を符号連結部69に出力 (供給) する。符号連結部69は、VLC符号化部62から供給された符号0を、ラインyの量子化係数の符号化の結果得られた符号として、そのまま出力する。

【 0 1 0 7 】

ステップS45において、最大有効桁数計算部63は、ライン判定部61からの量子化係数が全て0である旨の情報に基づいて、記憶している変数Binitの値をBinit = 0とし、変数Binitを更新する。

【 0 1 0 8 】

ステップS46において、ライン判定部61は、符号化しているサブバンドのラインのうち、未処理のラインがあるか否かを判定する。すなわち、ライン判定部61は、符号化しているサブバンドの全てのラインの量子化係数を符号化したか否かを判定する。例えば、図2に示したサブバンドの量子化係数を符号化している場合、ラインL1乃至ラインL6上の全ての位置の量子化係数が符号化されたとき、ライン判定部61は、未処理のラインが存在しないと判定する。

【 0 1 0 9 】

そのステップS46において、未処理のラインがあると判定した場合、ライン判定部61は、次のライン、すなわち、ライン (y + 1) 上の各位置の量子化係数を符号化するので、処理をステップS47に進める。

【 0 1 1 0 】

ステップS47において、ライン判定部61は、記憶しているラインを示す変数yをインクリメントしてy = y + 1とし、処理をステップS43に戻し、上述したそれ以降の処理を再び実行させる。

【 0 1 1 1 】

これに対して、ステップS46において、未処理のラインが無いと判定した場合、ライン判定部61は、サブバンドを構成する全てのラインについて量子化係数を符号化したので、エントロピ符号化処理を終了し、処理を図5のステップS13に戻し、符号化処理を終了させる。

【 0 1 1 2 】

また、図6のステップS43において、ラインyの量子化係数が全て0でない (0でない量子化係数が存在する) と判定した場合、ライン判定部61は、量子化係数が全て0でない (0でない量子化係数が存在する) 旨の情報を生成してそれをVLC符号化部62および最大有効桁

10

20

30

40

50

数計算部63に供給し、ステップS48に処理を進める。

【0113】

ステップS48において、VLC符号化部62は、ライン判定部61からの量子化係数が全て0でない旨の情報に基づいて、符号化するラインの量子化係数が全て0でないことを示す符号1を符号連結部69に出力（供給）する。

【0114】

ステップS49において、最大有効桁数計算部63は、ライン判定部61からの量子化係数が全て0でない旨の情報に基づいて、これから符号化するw個の量子化係数のうち、最初に入力される量子化係数に対応するラインy上の位置のx座標を示す変数xの値を $x=0$ として、この変数xを記憶する。

10

【0115】

例えば、ラインyが、図2に示したラインL1である場合、最大有効桁数計算部63が記憶している変数xの値は、これから符号化しようとするラインL1上の連続するw個の位置 $(x, 0)$ 、 $(x+1, 0)$ 、 $\dots$ 、 $(x+w-1, 0)$ のうちの図中、一番左側の位置 $(x, 0)$ のx座標を示している。

【0116】

また、ステップS49において、最大有効桁数計算部63は、前回符号化されたw個の量子化係数の最大有効桁数を示す変数Bの値を $B=B_{init}$ として、この変数Bを記憶する。すなわち、最大有効桁数計算部63は、変数Bの値を記憶している変数 $B_{init}$ の値として変数Bを更新し、更新された変数Bの値を記憶する。

20

【0117】

最大有効桁数計算部63は変数Bの値を更新すると、更新された変数B（最大有効桁数）の値を示す情報をVLC符号化部64および有効桁抽出部65に供給する。また、VLC符号化部64および有効桁抽出部65は、それぞれ最大有効桁数計算部63から供給された変数Bの値を記憶する。

【0118】

ステップS50において、エントロピ符号化部23はw個組符号化処理を行う。なお、w個組符号化処理の詳細は後述するが、w個組符号化処理において、エントロピ符号化部23は、ライン判定部61が記憶している変数yにより示されるラインy上の連続するw個の量子化係数を符号化する。

30

【0119】

ここで、ライン判定部61が記憶している変数yおよび、最大有効桁数計算部63が記憶している変数xにより特定されるラインy上の位置を $(x, y)$ とすると、ラインy上のw個の連続する位置は、ラインy上の連続する位置 $(x, y)$ 、 $(x+1, y)$ 、 $\dots$ 、 $(x+w-1, y)$ とされる。すなわち、w個組符号化処理において、エントロピ符号化部23は、位置 $(x, y)$ 、 $(x+1, y)$ 、 $\dots$ 、 $(x+w-1, y)$ のそれぞれの量子化係数を符号化する。

【0120】

ステップS51において、最大有効桁数計算部63は、ラインyに未処理の量子化係数があるか否かを判定する。すなわち、最大有効桁数計算部63は、ライン判定部61が記憶している変数yにより示されるラインy上の位置の全ての量子化係数を符号化したか否かを判定する。

40

【0121】

ステップS51において、ラインyに未処理の量子化係数があると判定した場合、最大有効桁数計算部63は、次のw個の量子化係数を符号化するので、処理をステップS52に進める。

【0122】

ステップS52において、最大有効桁数計算部63は、記憶している変数xを、 $x=x+w$ とし、処理をステップS50に戻す。これにより、その後のステップS50の処理において、ラインy上の位置 $(x+w, y)$ 、 $(x+w+1, y)$ 、 $\dots$ 、 $(x+2w-1, y)$ のそれぞれの量子化係数が符号化される。

【0123】

50

また、ステップS51において、ラインyに未処理の量子化係数がないと判定した場合、最大有効桁数計算部63は、ラインy上の一の全ての位置の量子化係数が符号化されたので、処理をステップS46に戻し、それ以降の処理を実行させる。

【0124】

このようにして、エントロピ符号化部23は、サブバンドの各位置の量子化係数を所定の数ずつラスタスキャン順に符号化する。

【0125】

このように、サブバンドの各位置の量子化係数を所定の数ずつラスタスキャン順に符号化することによって、入力された量子化係数を入力された順番に処理することができ、量子化係数の符号化により生じる遅延をより少なくすることができる。

10

【0126】

次に、図7のフローチャートを参照して、図6のステップS50の処理に対応するw個組符号化処理について説明する。

【0127】

ステップS81において、最大有効桁数計算部63は、記憶している変数xにより特定されるラインy上の位置を(x, y)として、連続するw個の位置(x, y)、(x+1, y)、・・・、(x+w-1, y)の量子化係数のうち、絶対値が最も大きい量子化係数の有効桁数を、これから符号化するw個の量子化係数の最大有効桁数を示す変数Bnewの値として、この変数Bnewを記憶する。

【0128】

20

また、最大有効桁数計算部63は、求められたw個の量子化係数の最大有効桁数、すなわち、変数Bnewの値をVLC符号化部64および有効桁抽出部65に供給する。

【0129】

例えば、w個の連続する位置の量子化係数のそれぞれが、図3に示した量子化係数“-0101”、“+0011”、“-0110”、および“+0010”である場合、これらの量子化係数のうち、絶対値が最大の量子化係数は“-0110”であり、その有効桁数は“-0110”の最上位にある1の桁である“3”であるので、変数Bnewの値は3とされる。

【0130】

ステップS82において、VLC符号化部64は、B=Bnewであるか否かを判定する。すなわち、VLC符号化部64は、記憶している、前回符号化されたw個の量子化係数の最大有効桁数を示す変数Bの値が、最大有効桁数計算部63から供給された、これから符号化するw個の量子化係数の最大有効桁数を示す変数Bnewの値と同じであるか否かを判定する。

30

【0131】

ステップS82において、B=Bnewであると判定した場合、VLC符号化部64は、処理をステップS83に進め、これから符号化するw個の量子化係数の最大有効桁数を示す符号として、最大有効桁数が変化していないことを示す符号0を符号連結部69に出力する。そして、最大有効桁数を示す符号0が出力すると、VLC符号化部64は、ステップS84の処理乃至ステップS88の処理をスキップし、ステップS89に処理を進める。

【0132】

これに対して、ステップS82において、B=Bnewでないと判定した場合、VLC符号化部64は、処理をステップS84に進め、(最大有効桁数が変化したので、)最大有効桁数が変化したことを示す符号1を符号連結部69に出力する。

40

【0133】

ステップS85において、VLC符号化部64は、次式(1)を満たす整数nおよびmを求める。

【0134】

$$B_{\text{new}} = B + (n + 1) \times (-1)^m \cdots (1)$$

【0135】

ここで、式(1)における記号“^”は、冪乗を表わしている。したがって、(-1)^mは、(-1)のm乗を表わしている。

【0136】

50

例えば、 $B_{\text{new}} = 3$ であり、 $B = 0$ である場合、式(1)を満たす $n$ および $m$ として、それぞれ $n = 2$ 、および $m = 0$ が求まる。変数 $B_{\text{new}}$ および変数 $B$ を比較した場合、変数 $B_{\text{new}}$ の値の絶対値と変数 $B$ の値の絶対値の差が大きいくほど、式(1)における $n$ の値は大きくなる。したがって $n$ の値は、最大有効桁数の変化量を示しているといえることができる。また、変数 $B_{\text{new}}$ の値が変数 $B$ の値よりも大きい場合、 $m$ の値は0となり、逆に変数 $B_{\text{new}}$ の値が変数 $B$ の値よりも小さい場合、 $m$ の値は1となる。したがって、式(1)における $m$ の値は、最大有効桁数が増加したか、または減少したかを示しているといえることができる。

【0137】

ステップS86において、VLC符号化部64は、最大有効桁数が増加したか、または減少したかを示す符号として、式(1)を満たす $m$ の値を1ビットの符号で符号連結部69に出力する。例えば、式(1)を満たす $m$ の値が0である場合、VLC符号化部64は、最大有効桁数が増加したことを示す符号0を出力する。

10

【0138】

ステップS87において、VLC符号化部64は、最大有効桁数の変化量を示す符号として、式(1)を満たす $n$ の値だけ連続する0に続けて1個の1を符号連結部69に出力する。すなわち、VLC符号化部64は、 $n$ 個の0および1個の1を最大有効桁数の変化量を示す符号として出力する。

【0139】

例えば、式(1)を満たす $n$ の値が2である場合、VLC符号化部64は、最大有効桁数の変化量を示す符号として“001”を符号連結部69に出力する。

20

【0140】

これにより、VLC符号化部64から符号連結部69には、これから符号化する $w$ 個の量子化係数の最大有効桁数を示す符号として、最大有効桁数が変化したことを示す符号、最大有効桁数が増加したか、または減少したかを示す符号、および最大有効桁数の変化量を示す符号が出力される。

【0141】

ステップS88において、最大有効桁数計算部63は、記憶している変数 $B$ の値を $B = B_{\text{new}}$ とし、処理をステップS89に進める。すなわち、最大有効桁数計算部63は、記憶している変数 $B$ の値を、記憶している変数 $B_{\text{new}}$ の値として変数 $B$ を更新する。また、VLC符号化部64および有効桁抽出部65も記憶している変数 $B$ の値を $B = B_{\text{new}}$ とする。

30

【0142】

ステップS88において、変数 $B$ の値が $B = B_{\text{new}}$ とされるか、またはステップS83において、量子化係数の最大有効桁数を示す符号が出力されると、ステップS89において、最大有効桁数計算部63は、記憶している変数 $x$ の値が0である場合、記憶している変数 $B_{\text{init}}$ の値を $B_{\text{init}} = B$ とする。

【0143】

すなわち、最大有効桁数計算部63は、記憶している変数 $x$ の値が0である場合、記憶している、ライン $(y - 1)$ 上の最初に入力される $w$ 個の量子化係数の最大有効桁数を示す変数 $B_{\text{init}}$ の値を、前回符号化された $w$ 個の量子化係数の最大有効桁数を示す変数 $B$ の値として、変数 $B_{\text{init}}$ を更新する。

40

【0144】

このように、変数 $x = 0$ である場合に、変数 $B_{\text{init}}$ の値を $B_{\text{init}} = B$ とすることで、次のライン(例えば、ライン $(y + 1)$ )の $x = 0$ から始まる $w$ 個の量子化係数について、前のライン(例えば、ライン $y$ )の $x = 0$ から始まる $w$ 個の量子化係数の最大有効桁数との相関関係を利用して量子化係数の符号化を行うことができる。

【0145】

ステップS90において、有効桁抽出部65は、所定の変数を $i$ として、変数 $i$ を0から $(w - 1)$ まで変化させて、量子化部22から供給された、ライン $y$ 上の位置 $(x + i, y)$ の量子化係数から、量子化係数の有効桁を抽出する。有効桁抽出部65は、抽出した量子化係数の有効桁(のデータ)をVLC符号化部66およびサイン抽出部67に供給する。また、VLC符号化部66

50

は、有効桁抽出部65から供給された有効桁に基づいて（有効桁を符号化して）、 $w$ 個の量子化係数の絶対値を示す符号を符号連結部69に出力する。

【0146】

ここで、位置 $(x+i, y)$ における $x$ の値は、最大有効桁数計算部63が記憶している変数 $x$ の値とされる。例えば、最大有効桁数計算部63が記憶している変数 $x$ の値が0であり、有効桁抽出部65が記憶している変数 $B$ の値が3であり、さらに、量子化部22から有効桁抽出部65に、位置 $(x+i, y)$  ( $0 \leq i \leq 3$ )のそれぞれ、すなわち位置 $(0, y)$ 、 $(1, y)$ 、 $(2, y)$ 、および $(3, y)$ の量子化係数に対応する図3に示した $w$ 個（4個）の量子化係数“-0101”、“+0011”、“-0110”、および“+0010”が供給されたとすると、有効桁抽出部65は、これらの量子化係数から有効桁を抽出する。

10

【0147】

この場合、有効桁抽出部65が記憶している変数 $B$ の値は3であり、有効桁は3桁であるから、有効桁抽出部65は、位置 $(x, y)$ に対応する量子化係数“-0101”から、最下位から3桁の値“101”を抽出する。

【0148】

同様に、有効桁抽出部65は、位置 $(x+1, y)$ 、位置 $(x+2, y)$ 、および位置 $(x+3, y)$ のそれぞれの量子化係数“+0011”、“-0110”、および“+0010”から、順次、これらの最下位から3桁の値“011”、“110”、および“010”を抽出する。これにより、有効桁抽出部65からVLC符号化部66およびサイン抽出部67には、量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の有効桁（の符号）“101”、“011”、“110”、および“010”が出力される。VLC符号化部66は、有効桁抽出部65から供給された符号“101”、“011”、“110”、および“010”を符号化して、 $w$ 個（4個）の量子化係数の絶対値を示す符号“101011110010”を、符号連結部69に出力する。

20

【0149】

ステップS91において、サイン抽出部67は、所定の変数を $i$ として、変数 $i$ を0から $(w-1)$ まで変化させて、量子化部22から供給された、量子化係数の絶対値が0でないライン $y$ 上の位置 $(x+i, y)$ の量子化係数から、量子化係数のサインを抽出し、抽出したサイン（のデータ）をVLC符号化部68に供給する。VLC符号化部68は、サイン抽出部67からのサインを符号化し、これにより得られる量子化係数のサインを示す符号を符号連結部69に出力する。

30

【0150】

符号連結部69は、VLC符号化部68から量子化係数のサインを示す符号が供給されると、VLC符号化部62、VLC符号化部64、VLC符号化部66、およびVLC符号化部68のそれぞれから供給された、量子化係数が全て0であるか否かを示す符号、量子化係数の最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号のそれぞれを連結して、連結された符号を符号化された画像として出力し、 $w$ 個組符号化処理を終了し、図6のステップS50に処理を戻し、ステップS51以降の処理を実行する。

【0151】

ここで、位置 $(x+i, y)$ における $x$ の値は、最大有効桁数計算部63が記憶している変数 $x$ の値とされる。例えば、最大有効桁数計算部63が記憶している変数 $x$ の値が0であり、量子化部22からサイン抽出部67に、位置 $(x+i, y)$  ( $0 \leq i \leq 3$ )のそれぞれ、すなわち位置 $(0, y)$ 、 $(1, y)$ 、 $(2, y)$ 、および $(3, y)$ の量子化係数に対応する図3に示した $w$ 個（4個）の量子化係数“-0101”、“+0011”、“-0110”、および“+0010”が供給されたとすると、量子化係数“-0101”、“+0011”、“-0110”、および“+0010”のそれぞれは0ではないので、サイン抽出部67は、これらの量子化係数からサインを抽出する。

40

【0152】

この場合、サイン抽出部67は、位置 $(x, y)$ に対応する量子化係数“-0101”から量子化係数のサイン“-”を抽出する。

【0153】

同様に、サイン抽出部67は、位置 $(x+1, y)$ 、位置 $(x+2, y)$ 、および位置 $(x+3, y)$ のそれぞれから量子化係数のサイン“-”を抽出する。

50



y)のそれぞれの量子化係数“+0011”、“-0110”、および“+0010”から、順次、これらの量子化係数のサイン“+”、“-”、および“+”を抽出する。これにより、サイン抽出部67からVLC符号化部68には、量子化係数“-0101”、“+0011”、“-0110”、および“+0010”のサイン“-”、“+”、“-”、および“+”が出力される。VLC符号化部68は、サイン抽出部67から供給された量子化係数のサイン“-”、“+”、“-”、および“+”を符号化する。

【0154】

例えば、VLC符号化部68は、サイン“-”が入力されると符号1を出力し、サイン“+”が入力されると符号0を出力することにより、入力されたサインを符号化する。この場合、VLC符号化部68には、量子化係数のサイン“-”、“+”、“-”、および“+”が入力されるので、VLC符号化部68は、符号“1”、“0”、“1”、“0”からなる符号“1010”を量子化係数のサインを示す符号として符号連結部69に出力する。

10

【0155】

このようにして、エントロピ符号化部23は、サブバンドの量子化係数を、予め定められた所定の数ずつまとめて符号化し、量子化係数の最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号を出力する。

【0156】

このように、サブバンドの量子化係数を、予め定められた所定の数ずつまとめて符号化することで、例えば、JPEG2000方式により画像を符号化する場合とは異なり、複数のコーディングパスに基づいて、ビットプレーンごとに複数の処理を行う必要がなく、また、可変長符号化を行うので、符号化の処理量を大幅に低減することができる。これにより、より高速に画像の符号化を行うことができ、高解像度の画像をリアルタイムで符号化するための符号化装置を安価で実現することができる。

20

【0157】

さらに、画像符号化装置11においては、画像を符号化する場合に、符号の長さを明示的に符号化する必要がないため、符号量を少なくすることができ、符号の長さの情報を管理する必要もない。

【0158】

なお、以上においては、w個の量子化係数のうち、絶対値の最も大きい量子化係数の有効桁数を、最大有効桁数を示す変数Bnewの値とすると説明したが、変数Bnewの値は、w個の量子化係数のうち、絶対値の最も大きい量子化係数の有効桁数以上の値であればよい。変数Bnewの値が大きくなると、量子化係数の絶対値を示す符号の符号量が多くなるが、変数Bnewの値を、絶対値の最も大きい量子化係数の有効桁数以上の値とすることで、量子化係数の最大有効桁数を示す符号の符号量を少なくすることができる。

30

【0159】

次に、画像符号化装置11により符号化された画像を復号する画像復号装置について説明する。

【0160】

図8は、画像復号装置の構成例を示すブロック図である。

【0161】

画像復号装置111は、エントロピ復号部121、逆量子化部122、およびウェーブレット逆変換部123から構成され、エントロピ復号部121には、符号化された画像（データ）が入力される。

40

【0162】

エントロピ復号部121は、入力された符号化された画像としての符号をエントロピ復号し、これにより得られた量子化係数を逆量子化部122に供給する。

【0163】

逆量子化部122は、エントロピ復号部121から供給された量子化係数を逆量子化し、逆量子化により得られた各サブバンドのウェーブレット係数をウェーブレット逆変換部123に供給する。

50

## 【 0 1 6 4 】

ウェーブレット逆変換部123は、逆量子化部122から供給された、各サブバンドのウェーブレット係数にウェーブレット逆変換を施し、その結果得られた画像を、復号された画像として出力する。

## 【 0 1 6 5 】

また、このような処理を行う画像復号装置111のエントロピ復号部121は、より詳細には、例えば、図9に示すように構成される。

## 【 0 1 6 6 】

エントロピ復号部121は、より詳細には、符号分割部151、ライン判定部152、発生部153、VLC復号部154、VLC復号部155、VLC復号部156、量子化係数合成部157、および切替部158を有する。

10

## 【 0 1 6 7 】

符号分割部151は、ライン判定部152、VLC復号部154、VLC復号部155、およびVLC復号部156のそれぞれから供給される情報に基づいて、入力された、符号化された画像としての符号を分割し、分割された所定の長さの符号をライン判定部152、VLC復号部154、VLC復号部155、またはVLC復号部156に供給する。

## 【 0 1 6 8 】

すなわち、符号分割部151は、入力された符号を、符号化された1つのラインの量子化係数が全て0であるか否かを示す符号、符号化されたw個の量子化係数の最大有効桁数を示す符号、符号化されたw個の量子化係数の絶対値を示す符号、および符号化された量子化係数のサインを示す符号のそれぞれに分割し、ライン判定部152、VLC復号部154、VLC復号部155、およびVLC復号部156のそれぞれに供給する。

20

## 【 0 1 6 9 】

ライン判定部152は、符号分割部151から供給された符号に基づいて、符号化されたサブバンドの1つのラインの量子化係数が全て0であるか否かを判定し、その判定の結果を示す情報を、符号分割部151、発生部153、およびVLC復号部154に供給する。

## 【 0 1 7 0 】

発生部153は、ライン判定部152からの判定の結果を示す情報に基づいて、1ライン分の0である量子化係数を示す符号を発生して切替部158に供給する。

## 【 0 1 7 1 】

30

VLC復号部154は、符号分割部151から供給された、符号化されたw個の量子化係数の最大有効桁数を示す符号を復号して、符号化されたw個の量子化係数の最大有効桁数を求め、求めた最大有効桁数を示す情報を符号分割部151、VLC復号部155、および量子化係数合成部157に供給する。

## 【 0 1 7 2 】

VLC復号部155は、VLC復号部154からの最大有効桁数を示す情報に基づいて、符号分割部151から供給された量子化係数の絶対値を示す符号を復号し、これにより得られたw個の量子化係数の有効桁（のデータ）を、VLC復号部156および量子化係数合成部157に供給する。また、VLC復号部155は、量子化係数の絶対値を示す符号の復号の結果を示す情報を符号分割部151に供給する。

40

## 【 0 1 7 3 】

VLC復号部156は、VLC復号部155から供給された量子化係数の有効桁に基づいて、符号分割部151から供給された量子化係数のサインを示す符号を復号し、これにより得られる量子化係数のサイン（のデータ）を量子化係数合成部157に供給する。また、VLC復号部156は、量子化係数のサインを示す符号の復号の結果を示す情報を符号分割部151に供給する。

## 【 0 1 7 4 】

量子化係数合成部157は、VLC復号部154からの最大有効桁数を示す情報に基づいて、VLC復号部155から供給された量子化係数の有効桁およびVLC復号部156から供給された量子化係数のサインを合成し、これにより得られたw個の量子化係数を切替部158に供給する。

50

## 【 0 1 7 5 】

切替部158は、発生部153または量子化係数合成部157からの量子化係数を出力する。

## 【 0 1 7 6 】

図10は、符号分割部151のより詳細な構成例を示すブロック図である。

## 【 0 1 7 7 】

符号分割部151は、制御部171およびメモリ172を有している。制御部171は、符号化された画像としての符号が入力されると、入力された符号をメモリ172に供給して一時的に記憶させる。

## 【 0 1 7 8 】

そして、制御部171は、図9に示したライン判定部152、VLC復号部154、VLC復号部155、およびVLC復号部156のそれぞれから供給される情報に基づいて、メモリ172に一時的に記憶されている符号のうち、所定の長さの符号を読み出して、ライン判定部152、VLC復号部154、VLC復号部155、またはVLC復号部156に供給する。

10

## 【 0 1 7 9 】

また、符号分割部151は、図10に示した構成例の他、例えば、図11に示すように構成されてもよい。

## 【 0 1 8 0 】

図11に示す符号分割部151は、制御部191、スイッチ192、並びに、ノード193-1乃至ノード193-4を有する。

## 【 0 1 8 1 】

制御部191は、符号化された画像としての符号が符号分割部151に入力されると、図9に示したライン判定部152、VLC復号部154、VLC復号部155、およびVLC復号部156のそれぞれから供給される情報に基づいて、スイッチ192を制御し、入力された符号のうち、所定の長さの符号をライン判定部152、VLC復号部154、VLC復号部155、またはVLC復号部156に供給させる。

20

## 【 0 1 8 2 】

すなわち、ノード193-1乃至ノード193-4のそれぞれは、ライン判定部152、VLC復号部154、VLC復号部155、およびVLC復号部156のそれぞれに接続されており、制御部191は、符号の供給先として、ノード193-1乃至ノード193-4のいずれかを選択し、スイッチ192と、選択されたノードとの接続を制御する。

30

## 【 0 1 8 3 】

スイッチ192が制御部191の制御に基づいて選択されたノードを入力と接続するので、符号分割部151に入力された符号は、スイッチ192、およびスイッチ192と接続されたノードを介して、符号の供給先として選択されたライン判定部152、VLC復号部154、VLC復号部155、またはVLC復号部156に供給される。

## 【 0 1 8 4 】

次に、図12のフローチャートを参照して、画像復号装置111による復号処理について説明する。この復号処理は、エントロピ復号部121に符号化された画像としての符号が入力されると開始される。

## 【 0 1 8 5 】

ステップS131において、エントロピ復号部121は、エントロピ復号処理を行い、入力された画像としての符号をエントロピ復号し、これにより得られた量子化係数を逆量子化部122に供給する。なお、エントロピ復号処理の詳細は後述するが、このエントロピ復号処理において、エントロピ復号部121は、符号化されたサブバンドのライン上の連続する位置の量子化係数をw個ずつ復号し、復号された量子化係数を逆量子化部122に供給する。

40

## 【 0 1 8 6 】

ステップS132において、逆量子化部122は、エントロピ復号部121から供給された量子化係数を逆量子化し、逆量子化により得られた各サブバンドのウェーブレット係数をウェーブレット逆変換部123に供給する。

## 【 0 1 8 7 】

50

ステップS133において、ウェーブレット逆変換部123は、逆量子化部122から供給された、各サブバンドのウェーブレット係数にウェーブレット逆変換を施し、その結果得られた画像を出力して復号処理は終了する。

【0188】

このようにして、画像復号装置111は、符号化された画像を復号して出力する。

【0189】

次に、図13のフローチャートを参照して、図12のステップS131の処理に対応するエントリピ復号処理について説明する。

【0190】

ステップS161において、ライン判定部152は、これから復号するサブバンドのラインを示す変数 $y$ を $y = 0$ として、これを記憶する。

10

【0191】

ステップS162において、VLC復号部154は、ライン判定部152が記憶している変数 $y$ により示されるライン $y$ よりも1つ前のライン( $y - 1$ )上の最初に入力される $w$ 個の量子化係数の最大有効桁数を示す変数 $Binit$ を $Binit = 0$ として、これを記憶する。

【0192】

例えば、ライン( $y - 1$ )が、図2に示したラインL1である場合、ライン( $y - 1$ )上の最初に入力される $w$ 個の量子化係数の最大有効桁数を示す変数 $Binit$ の値は、ラインL1の図中、左端の位置から $w$ 個の量子化係数の最大有効桁数となる。また、ライン判定部152が記憶している変数 $y$ が $y = 0$ である場合、ライン( $y - 1$ )は存在しないので、変数 $Binit$ の値は、 $Binit = 0$ とされる。

20

【0193】

また、ステップS162において、符号分割部151は、入力された符号のうち、最初の1ビットの符号を、これから復号するラインの量子化係数が全て0であるか否かを示す符号として、ライン判定部152に供給する。

【0194】

ステップS163において、ライン判定部152は、符号分割部151から読み込んだ(供給された)1ビットの符号が0であるか否かを判定し、その判定の結果を示す情報を生成して発生部153、VLC復号部154、および符号分割部151に供給する。

【0195】

30

ステップS163において、符号が0であると判定した場合、ライン $y$ の量子化係数が全て0であるので、ライン判定部152は、処理をステップS164に進める。ステップS164において発生部153は、ライン判定部152からの判定の結果を示す情報に基づいて、ライン $y$ の量子化係数を全て0とする。そして、発生部153は、ライン $y$ の量子化係数を示す符号を発生して切替部158に供給する。

【0196】

例えば、図3に示したように、1つの量子化係数が4桁で表わされ、1つのラインの量子化係数が5個である場合、発生部153は、ライン $y$ の量子化係数を示す符号として、 $20 (= 4 \times 5)$ 個の0を発生して切替部158に供給する。切替部158は、発生部153から供給された連続する20個の0を、1つのラインの量子化係数を示す符号として、逆量子化部122に出力する。

40

【0197】

ステップS165において、VLC復号部154は、ライン判定部152からの判定の結果を示す情報に基づいて、記憶している変数 $Binit$ の値を $Binit = 0$ とし、変数 $Binit$ を更新する。

【0198】

ステップS166において、ライン判定部152は、復号しているサブバンドのラインのうち、未処理のラインがあるか否かを判定する。すなわち、ライン判定部152は、復号しているサブバンドの全てのライン上の位置の量子化係数を復号したか否かを判定する。

【0199】

ステップS166において、未処理のラインがあると判定した場合、ライン判定部152は、

50

自分自身が記憶している変数 $y$ により示されるライン $y$ の次のライン( $y+1$ )上の各位置の量子化係数を復号するので、処理をステップS167に進める。

【0200】

ステップS167において、ライン判定部152は、記憶しているラインを示す変数 $y$ をインクリメントして $y=y+1$ とし、処理をステップS163に戻し、それ以降の処理を実行させる。

【0201】

これに対して、ステップS166において、未処理のラインが無いと判定した場合、サブバンドを構成する全てのラインについて量子化係数が復号されたので、ライン判定部152は、エントロピ復号処理を終了し、処理を図12のステップS131に戻し、ステップS132以降の処理を実行させる。

【0202】

また、図13のステップS163において、符号が0でないと判定した場合、ライン判定部152は、処理をステップS168に進める。ステップS168において、VLC復号部154は、ライン判定部152からの判定の結果を示す情報に基づいて、これから復号する $w$ 個の量子化係数のうち、最初に入力される量子化係数に対応するライン $y$ 上の位置の $x$ 座標を示す変数 $x$ の値を $x=0$ として、この変数 $x$ を記憶する。

【0203】

また、ステップS168において、VLC復号部154は、前回復号された $w$ 個の量子化係数の最大有効桁数を示す変数 $B$ の値を $B=B_{init}$ として、この変数 $B$ を記憶する。すなわち、VLC復号部154は、変数 $B$ の値を記憶している変数 $B_{init}$ の値として変数 $B$ を更新し、更新された変数 $B$ の値を記憶する。

【0204】

さらに、ステップS168において、符号分割部151は、ライン判定部152からの判定の結果を示す情報に基づいて、入力された符号の次の1ビットの符号を、これから復号する $w$ 個の量子化係数の最大有効桁数が変化したか否かを示す符号として、VLC復号部154に供給する。

【0205】

ステップS169において、エントロピ復号部121は、 $w$ 個組復号処理を行う。なお、 $w$ 個組復号処理の詳細は後述するが、この $w$ 個組復号処理において、エントロピ復号部121は、ライン判定部152が記憶している変数 $y$ により示されるライン $y$ 上の連続する $w$ 個の位置の量子化係数を復号する。

【0206】

ステップS170において、VLC復号部154は、ライン $y$ に未処理の量子化係数があるか否かを判定する。すなわち、VLC復号部154は、ライン判定部152が記憶している変数 $y$ により示されるライン $y$ 上の位置の全ての量子化係数を復号したか否かを判定する。

【0207】

ステップS170において、ライン $y$ に未処理の量子化係数があると判定した場合、次の $w$ 個の量子化係数を復号するので、VLC復号部154は、処理をステップS171に進める。

【0208】

ステップS171において、VLC復号部154は、記憶している変数 $x$ を、 $x=x+w$ として、処理をステップS169に戻す。これにより、その後のステップS169の処理において、ライン $y$ 上の位置 $(x+w, y)$ 、 $(x+w+1, y)$ 、 $\dots$ 、 $(x+2w-1, y)$ のそれぞれの量子化係数が復号される。

【0209】

また、ステップS170において、ライン $y$ に未処理の量子化係数がないと判定した場合、VLC復号部154は、ライン $y$ 上の全ての位置の量子化係数が復号されたので、処理をステップS166に戻し、それ以降の処理を実行させる。

【0210】

このようにして、エントロピ復号部121は、サブバンドの各位置の量子化係数を所定の数ずつラスタスキャン順に復号する。

10

20

30

40

50

## 【 0 2 1 1 】

このように、サブバンドの各位置の量子化係数を所定の数ずつラスタスキャン順に復号することによって、符号化された量子化係数を、入力された順番に処理することができ、量子化係数の復号により生じる遅延をより少なくすることができる。

## 【 0 2 1 2 】

次に、図14のフローチャートを参照して、図13のステップS169の処理に対応するw個組復号処理について説明する。

## 【 0 2 1 3 】

上述したように、図13のステップS168において、符号分割部151からVLC復号部154には、これから復号するw個の量子化係数の最大有効桁数が変化したか否かを示す1ビットの符号が供給される。

10

## 【 0 2 1 4 】

図14のステップS211において、VLC復号部154は、読み込んだ（供給された）1ビットの符号が0であるか否かを判定する。

## 【 0 2 1 5 】

ステップS211において、読み込んだ符号が0であると判定した場合、最大有効桁数は変化していないので、VLC復号部154は、最大有効桁数が変化していない旨の情報を生成し、それを符号分割部151、VLC復号部155、および量子化係数合成部157に供給し、ステップS212乃至ステップS214の各処理をスキップして、ステップS215に処理を進める。

20

## 【 0 2 1 6 】

すなわち、最大有効桁数が変化したか否かを示す符号が0である場合、図3を参照して説明したように、最大有効桁数が変化したか否かを示す1ビットの符号0の後には、最大有効桁数が増加したか、または減少したかを示す符号、および最大有効桁数の変化量を示す符号ではなく、量子化係数の絶対値を示す符号が入力されるので、最大有効桁数が増加したか、または減少したかを示す符号、および最大有効桁数の変化量を示す符号を復号する処理であるステップS212乃至ステップS214の各処理はスキップされる。

## 【 0 2 1 7 】

これに対して、ステップS211において、読み込んだ1ビットの符号が0でないとして判定した場合、最大有効桁数が変化したので、VLC復号部154は、処理をステップS212に進め、符号分割部151から符号を1ビット読み込み、その値を所定の変数mとして記憶する。

30

## 【 0 2 1 8 】

ステップS213において、VLC復号部154は、符号が1となるまで（符号1を読み込むまで）符号分割部151から符号を読み込み、それまでに読み込んだ符号0の数を所定の変数nとして記憶する。例えば、VLC復号部154が符号分割部151から読み込んだ3つ目の符号が1であった場合、すなわち、VLC復号部154が符号“001”を読み込んだ場合、VLC復号部154が符号1を読み込むまでに、読み込んだ符号0の数は2であるので、VLC復号部154は、読み込んだ符号0の数である2を変数nの値として記憶する。

## 【 0 2 1 9 】

ステップS214において、VLC復号部154は、最大有効桁数を示す変数Bの値を以下の式（2）により求め、求められた変数Bの値を記憶する。

40

## 【 0 2 2 0 】

$$B = B + (n + 1) \times (-1)^m \quad \cdots (2)$$

## 【 0 2 2 1 】

ここで、式（2）における左辺は、新たに求められる変数Bの値を表わしており、右辺のBは記憶している変数Bの値を表わしている。また、式（2）における記号“ $\wedge$ ”は、冪乗を表わしている。したがって、 $(-1)^m$ は、 $(-1)$ のm乗を表わしている。

## 【 0 2 2 2 】

VLC復号部154は、記憶している変数B、変数m、および変数nに基づいて、式（2）を計算して記憶している変数Bを更新する。VLC復号部154は、最大有効桁数を示す変数Bを更新すると、更新された最大有効桁数を示す情報を生成して、符号分割部151、VLC復号部155、

50

および量子化係数合成部157に供給する。

【0223】

ステップS214において新たな最大有効桁数を求めるか、ステップS211において、読み込んだ1ビットの符号が0であると判定すると、VLC復号部154は、ステップS215に処理を進め、記憶している変数 $x$ の値が0である場合、記憶している変数 $B_{init}$ の値を $B_{init} = B$ とする。

【0224】

すなわち、VLC復号部154は、記憶している変数 $x$ の値が0である場合、記憶している、ライン $(y - 1)$ 上の最初に入力される $w$ 個の量子化係数の最大有効桁数を示す変数 $B_{init}$ の値を、これから復号する $w$ 個の量子化係数の最大有効桁数を示す変数 $B$ の値として、変数 $B_{init}$ を更新する。

10

【0225】

このように、変数 $x = 0$ である場合に、変数 $B_{init}$ の値を $B_{init} = B$ とすることで、次のライン（例えば、ライン $(y + 1)$ ）の $x = 0$ から始まる $w$ 個の量子化係数について、前のライン（例えば、ライン $y$ ）の $x = 0$ から始まる $w$ 個の量子化係数の最大有効桁数との相関関係を利用して量子化係数の復号を行うことができる。

【0226】

ステップS216において、VLC復号部155は、所定の変数を $i$ として、変数 $i$ を0から $(w - 1)$ まで変化させて、符号分割部151から符号を $B$ ビットずつ読み込み、読み込んだ $B$ ビットの符号を、ライン $y$ 上の位置 $(x + i, y)$ の量子化係数の有効桁を示す符号として、VLC復号部156および量子化係数合成部157に供給（出力）する。また、VLC復号部155は、量子化係数の有効桁を示す情報を生成して、符号分割部151に供給する。

20

【0227】

ここで、位置 $(x + i, y)$ における $x$ の値は、VLC復号部154が記憶している変数 $x$ の値とされる。例えば、VLC復号部154が記憶している変数 $x$ の値が0であり、VLC復号部155が記憶している変数 $B$ の値が3である場合、VLC復号部155は、変数 $i = 0$ として符号分割部151から3ビットの符号を読み込み、読み込んだ3ビットの符号を位置 $(0, y)$ における量子化係数の有効桁として出力する。

【0228】

同様に、VLC復号部155は、変数 $i = 1$ として符号分割部151からさらに3ビットの符号を読み込み、その符号を位置 $(1, y)$ における量子化係数の有効桁として出力し、変数 $i = 2$ として符号分割部151から次の3ビットの符号を読み込み、その符号を位置 $(2, y)$ における量子化係数の有効桁として出力し、変数 $i = 3$ として符号分割部151から、さらに次の3ビットの符号を読み込み、その符号を位置 $(3, y)$ における量子化係数の有効桁として出力する。

30

【0229】

ステップS217において、VLC復号部156は、所定の変数を $i$ として、変数 $i$ を0から $(w - 1)$ まで変化させて、ライン $y$ 上の位置 $(x + i, y)$ の量子化係数の有効桁（絶対値）が0でなければ、符号分割部151から符号を1ビット読み込む。そして、VLC復号部156は、読み込んだ符号を復号し、これにより得られた符号を量子化係数のサインとして量子化係数合成部157に供給（出力）する。また、VLC復号部156は、量子化係数のサインを示す情報を生成して、符号分割部151に供給する。

40

【0230】

ここで、位置 $(x + i, y)$ における $x$ の値は、VLC復号部154が記憶している変数 $x$ の値とされる。例えば、VLC復号部154が記憶している変数 $x$ の値が0であり、VLC復号部155から0でない有効桁（を示す符号）が供給された場合、VLC復号部156は、変数 $i = 0$ として符号分割部151から1ビットの符号を読み込み、その符号が0であれば、位置 $(0, y)$ における量子化係数のサイン“-”を示す符号を量子化係数合成部157に供給し、読み込んだ符号が1であれば、位置 $(0, y)$ における量子化係数のサイン“+”を示す符号を量子化係数合成部157に供給する。

50

## 【 0 2 3 1 】

また、VLC復号部156は、VLC復号部155から供給された有効桁（を示す符号）の絶対値が0である場合、位置（0, y）における量子化係数のサインはないので、符号分割部151から符号を読み込まない。

## 【 0 2 3 2 】

同様にして、VLC復号部155から次に供給された有効桁（の絶対値）が0でない場合、VLC復号部156は、変数  $i = 1$  として符号分割部151から1ビットの符号を読み込み、その符号が0であれば、サイン“-”を示す符号を、読み込んだ符号が1であれば、サイン“+”を示す符号を量子化係数合成部157に供給する。

## 【 0 2 3 3 】

さらに、VLC復号部155から次に供給された有効桁が0でない場合、VLC復号部156は、変数  $i = 2$  として符号分割部151から1ビットの符号を読み込み、その符号が0であれば、サイン“-”を示す符号を、読み込んだ符号が1であれば、サイン“+”を示す符号を量子化係数合成部157に供給する。また、VLC復号部155から次に供給された有効桁が0でない場合、VLC復号部156は、変数  $i = 3$  として符号分割部151から1ビットの符号を読み込み、その符号が0であれば、サイン“-”を示す符号を、読み込んだ符号が1であれば、サイン“+”を示す符号を量子化係数合成部157に供給する。

## 【 0 2 3 4 】

ステップS218において、量子化係数合成部157は、VLC復号部155から供給された有効桁、およびVLC復号部155から供給されたサインを合成し、これにより得られた量子化係数を、切替部158を介して逆量子化部122に出力し、w個組復号処理を終了して処理を図13のステップS169に戻し、ステップS170以降の処理を実行させる。

## 【 0 2 3 5 】

例えば、出力する量子化係数の絶対値の桁数は予め定められている。予め定められた出力する量子化係数の絶対値の桁数が4桁であり、VLC復号部154からの最大有効桁数を示す情報により示される最大有効桁数が3である場合、VLC復号部155から有効桁“101”が供給され、VLC復号部155からサイン“-”を示す符号が供給されたとき、量子化係数合成部157は、量子化係数“-0101”を出力する。

## 【 0 2 3 6 】

すなわち、量子化係数合成部157は、量子化係数の絶対値の桁数は4桁であり、有効桁は“101”（3桁）であるので、有効桁“101”の最上位のビットのさらに1つ上位のビットを0として、量子化係数の絶対値を4桁である“0101”とする。そして、さらに、量子化係数のサイン“-”と、量子化係数の絶対値“0101”とを合成して得られる“-0101”を量子化係数として出力する。

## 【 0 2 3 7 】

なお、量子化係数合成部157は、VLC復号部155から供給された有効桁が0である場合には、サインのない量子化係数を出力する。例えば、予め定められた出力する量子化係数の絶対値の桁数が4桁であり、VLC復号部154からの最大有効桁数を示す情報により示される最大有効桁数が3である場合、VLC復号部155から有効桁“000”が供給されたとき、量子化係数合成部157は、量子化係数“0000”を出力する。

## 【 0 2 3 8 】

このようにして、エン트로ピ復号部121は、符号化されたサブバンドの量子化係数を、予め定められた所定の数ずつまとめて復号する。

## 【 0 2 3 9 】

このように、符号化されたサブバンドの量子化係数を、予め定められた所定の数ずつまとめて復号することで、例えば、JPEG2000方式により画像を復号する場合とは異なり、複数のコーディングパスに基づいて、ビットプレーンごとに複数の処理を行う必要がなく、より高速に画像の復号を行うことができる。これにより、高解像度の画像をリアルタイムで復号するための復号装置を安価で実現することができる。

## 【 0 2 4 0 】



ところで、以上において説明した画像符号化装置11においては、量子化係数の絶対値を符号化（または復号）する場合に、予め定められたw個の量子化係数の絶対値を順番に符号化すると説明したが、汎用DSP（Digital Signal Processor）や、汎用CPUにおいて用いられているSIMD（Single Instruction Multiple Data）演算命令を利用して、w個の量子化係数を同時に（並列に）符号化（または復号）することにより、より高速に画像を符号化（または復号）することができる。

【0241】

ここで、SIMD演算命令の一例としては、例えば、インテル株式会社のCPUにおいて用いられているMMX（MultiMedia eXtension）、SSE（Streaming SIMD Extensions）、SSE2、SSE3などがある。

10

【0242】

このようなSIMD演算命令を利用して量子化係数の絶対値を符号化する場合、画像符号化装置11のエントロピ符号化部23は、例えば、図15に示すように構成される。

【0243】

図15に示すエントロピ符号化部23は、ライン判定部61、VLC符号化部62、最大有効桁数計算部63、VLC符号化部64、有効桁抽出部65、VLC符号化部66、サイン抽出部67、およびVLC符号化部68が設けられている点において、図4に示したエントロピ符号化部23と同様であり、符号連結部69に新たにバッファ301が設けられている点において異なる。なお、図15において、図4における場合と対応する部分については同一の符号を付してあり、その説明は適宜省略する。

20

【0244】

符号連結部69のバッファ301には、VLC符号化部62、VLC符号化部64、VLC符号化部66、およびVLC符号化部68のそれぞれから供給された、ラインの量子化係数が全て0であるか否かを示す符号、最大有効桁数を示す符号、量子化係数の絶対値を示す符号、および量子化係数のサインを示す符号のそれぞれが一時的に記憶される。

【0245】

バッファ301の記憶領域は、32ビットごとに管理され、バッファ301に入力される符号（データ）は、スカラ演算処理に用いられる符号と、ベクトル演算処理に用いられる符号とに分けられて記憶される。すなわち、1つの32ビットの記憶領域には、一時的に記憶する符号（データ）として、スカラ演算処理に用いられる符号、またはベクトル演算処理に用いられる符号が記憶される。

30

【0246】

図15のエントロピ符号化部23においては、量子化係数の絶対値がSIMD演算命令を利用して並列に符号化されるので、量子化係数の絶対値を示す符号がベクトル演算処理に用いられる符号とされ、その他の符号は、スカラ演算処理に用いられる符号とされる。

【0247】

なお、以下の説明においては、バッファ301に設けられる32ビットの記憶領域のうち、スカラ演算処理に用いられる符号が記憶される記憶領域をスカラ領域とも称し、ベクトル演算処理に用いられる符号が記憶される記憶領域をベクトル領域とも称する。

【0248】

40

次に、図16を参照して、図15のエントロピ符号化部23が行うエントロピ符号化について説明する。

【0249】

例えば、図16の左上に示すように、図3に示した12個の量子化係数“-0101”、“+0011”、“-0110”、“+0010”、“+0011”、“+0110”、“0000”、“-0011”、“+1101”、“-0100”、“+0111”、および“-1010”がエントロピ符号化部23に順番に入力されたとする。

【0250】

すると、図3を参照して説明した場合と同様に、エントロピ符号化部23の符号連結部69には、これから符号化するラインの量子化係数が全て0であるか否かを示す符号“1”、お

50

よび最初に入力された4つの量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の最大有効桁数を示す符号“10001”が供給される。

【0251】

そして、これらの量子化係数が全て0であるか否かを示す符号“1”、および量子化係数の最大有効桁数を示す符号“10001”からなる符号“110001”は、矢印A11に示すように、符号連結部69のバッファ301に設けられた32ビットのスカラー領域に記憶される。

【0252】

図16の例においては、スカラー領域は、さらに4つの8ビットの領域に分けられており、スカラー領域には、その領域に記憶される符号が図中、左から右方向に、上位のビットから順番に格納されて記憶される。そして、1つのスカラー領域の全体に符号が格納されると、すなわち、1つのスカラー領域に32ビットの符号が格納されると、バッファ301には、新たな1つのスカラー領域が設けられ、新たに設けられたスカラー領域に、順次スカラー演算処理に用いられる符号が格納されていく。

10

【0253】

量子化係数が全て0であるか否かを示す符号“1”、および量子化係数の最大有効桁数を示す符号“10001”からなる符号“110001”がスカラー領域に記憶されると、次に、エン트로ピ符号化部23は、最初に入力されたw個(4個)の量子化係数“-0101”、“+0011”、“-0110”、および“+0010”のそれぞれの絶対値を示す最大有効桁数分の符号を、同時に(並列に並べて)ベクトル領域に記憶する。

【0254】

20

量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の最大有効桁数は、図3を参照して説明したように、“3”であるから、これらの4個の量子化係数の絶対値を示す符号は、それぞれ“101”、“011”、“110”、および“010”となり、矢印A12に示すように、量子化係数の絶対値を示す符号“101”、“011”、“110”、および“010”は、バッファ301に設けられた1つのベクトル領域に並列に並べられて格納され、記憶される。

【0255】

ここで、ベクトル領域は、さらに4つの8ビットの領域に分けられており、ベクトル領域の4つの領域のそれぞれには、同じ長さの(ビット長の)4つの量子化係数の絶対値を示す符号のそれぞれが、図中、左から右方向に、上位のビットから順番に格納されて記憶される。

30

【0256】

矢印A12により示されるベクトル領域においては、図中、左側の8ビットの領域に量子化係数の絶対値を示す符号“101”が左側から格納され、左から2番目の8ビットの領域に符号“011”が左側から格納され、右から2番目の領域に符号“110”が左側から格納され、さらに、一番右側の領域には符号“010”が左側から格納されて記憶されている。

【0257】

また、ベクトル領域もスカラー領域における場合と同様に、1つのベクトル領域の全体に符号が格納されると、すなわち、1つのベクトル領域に32ビットの符号が格納されると、バッファ301には、新たな1つのベクトル領域が設けられ、新たに設けられたベクトル領域に、順次ベクトル演算処理に用いられる符号が格納されていく。

40

【0258】

そして、4つの量子化係数“-0101”、“+0011”、“-0110”、および“+0010”の絶対値を示す符号がベクトル領域に記憶されると、エン트로ピ符号化部23は、矢印A13に示すように、これらの4つの量子化係数のサインを示す符号をスカラー領域に格納して記憶する。

【0259】

矢印A11に示したようにスカラー領域には、量子化係数が全て0であるか否かを示す符号“1”、および量子化係数の最大有効桁数を示す符号“10001”からなる符号“110001”が既に記憶されているので、エン트로ピ符号化部23は、矢印A13に示すように、量子化係数“-0101”、“+0011”、“-0110”、および“+0010”のサインを示す符号“1010”を、スカ

50

ラ領域の既に記憶されている符号“110001”の右側に(符号“110001”に続けて)格納して記憶する。

【0260】

さらに、最初の4つの量子化係数が符号化されると、エントロピ符号化部23は、次の4つの量子化係数“+0011”、“+0110”、“0000”、および“-0011”を符号化する。

【0261】

まず、エントロピ符号化部23は、前回符号化した4つの量子化係数の最大有効桁数“3”と、今回符号化する量子化係数“+0011”、“+0110”、“0000”、および“-0011”の最大有効桁数“3”とを比較し、最大有効桁数が変化していないので、矢印A14に示すように、最大有効桁数を示す符号として、最大有効桁数が変化していないことを示す符号“0”を、既に記憶されている符号“1100011010”に続けてスカラ領域に格納して記憶する。

10

【0262】

次に、エントロピ符号化部23は、今回入力されたw個(4個)の量子化係数“+0011”、“+0110”、“0000”、および“-0011”のそれぞれの絶対値を示す最大有効桁数分の符号“011”、“110”、“000”、および“011”のそれぞれを、矢印A15に示すようにベクトル領域に同時に格納して記憶する。

【0263】

矢印A12に示したように、ベクトル領域の図中、左側の8ビットの領域、左から2番目の8ビットの領域、右から2番目の8ビットの領域、および一番右側の8ビットの領域のそれぞれには、既に符号“101”、“011”、“110”、および“010”のそれぞれが記憶されているので、エントロピ符号化部23は、矢印A15に示すように、今回入力された量子化係数の絶対値を示す符号“011”、“110”、“000”、および“011”のそれぞれを、ベクトル領域に既に記憶されている符号“101”、“011”、“110”、および“010”のそれぞれの右側に続けて格納して記憶する。

20

【0264】

さらに、エントロピ符号化部23は、矢印A16に示すように、今回入力された4つの量子化係数“+0011”、“+0110”、“0000”、および“-0011”のうち、絶対値が0でない量子化係数のサインを示す符号“001”を、スカラ領域内の既に記憶されている符号“11000110100”の右側に続けて格納して記憶する。

【0265】

そして、4つの量子化係数“+0011”、“+0110”、“0000”、および“-0011”が符号化されると、エントロピ符号化部23は、次の4つの量子化係数“+1101”、“-0100”、“+0111”、および“-1010”の符号化を行う。

30

【0266】

まず、エントロピ符号化部23は、今回入力された4つの量子化係数“+1101”、“-0100”、“+0111”、および“-1010”の最大有効桁数“4”と、前回符号化した4つの量子化係数“3”とを比較して、矢印A17に示すように、最大有効桁数が変化したことを示す符号“1”、最大有効桁数が増加したことを示す符号“0”、および最大有効桁数の変化量を示す符号“1”からなる、最大有効桁数を示す符号“101”をスカラ領域に格納して記憶する。

【0267】

この場合、スカラ領域には、矢印A16に示したように、既に符号“11000110100001”が記憶されているので、エントロピ符号化部23は、矢印A17に示すように、この符号“11000110100001”の図中、右側に最大有効桁数を示す符号“101”を格納する。

40

【0268】

さらに、4つの量子化係数“+1101”、“-0100”、“+0111”、および“-1010”の最大有効桁数を示す符号が記憶されると、エントロピ符号化部23は、これらの量子化係数の絶対値を示す符号“1101”、“0100”、“0111”、および“1010”のそれぞれを、矢印A18に示すようにベクトル領域に同時に格納して記憶する。

【0269】

矢印A15に示したように、ベクトル領域の図中、左側の8ビットの領域、左から2番目の8

50

ビットの領域、右から2番目の8ビットの領域、および一番右側の8ビットの領域のそれぞれには、既に符号“101011”、“011110”、“110000”、および“010011”のそれぞれが記憶されており、左側の8ビットの領域、左から2番目の8ビットの領域、右から2番目の8ビットの領域、および一番右側の8ビットの領域のそれぞれには、それぞれ2ビットの符号しか記憶することができない。

【0270】

そこで、エントロピ符号化部23は、矢印A18に示すように、バッファ301に新たなベクトル領域を確保し（設け）、今回入力された量子化係数の絶対値を示す符号“1101”、“0100”、“0111”、および“1010”のそれぞれのうちの上位の2ビットの符号“11”、“01”、“01”、および“10”のそれぞれを、ベクトル領域に既に記憶されている符号“101011”、“011110”、“110000”、および“010011”のそれぞれの右側に続けて格納して記憶し、今回入力された量子化係数の絶対値を示す符号“1101”、“0100”、“0111”、および“1010”のそれぞれのうちの下位の2ビットの符号“01”、“00”、“11”、および“10”のそれぞれを、新たに設けられたベクトル領域（矢印A18により示される2つのベクトル領域のうち、図中、下側のベクトル領域）の図中、左側の8ビットの領域、左から2番目の8ビットの領域、右から2番目の8ビットの領域、および一番右側の8ビットの領域のそれぞれの左側に格納して記憶する。

10

【0271】

そして、4つの量子化係数“+1101”、“-0100”、“+0111”、および“-1010”の絶対値を示す符号が記憶されると、矢印A19に示すように、エントロピ符号化部23は、これらの4つの量子化係数のうち、絶対値が0でない量子化係数のサインを示す符号“0101”を、スカラ領域内の既に記憶されている符号“11000110100001101”の右側に続けて格納して記憶する。

20

【0272】

このようにして、入力された量子化係数が符号化されると、エントロピ符号化部23は、矢印A19により示されるスカラ領域に記憶されている符号、矢印A19により示される2つのベクトル領域のうち、図中、上側のベクトル領域に記憶されている符号、および下側のベクトル領域に記憶されている符号を、符号化された画像として順番に出力する。

【0273】

この場合、矢印A19により示されるスカラ領域の図中、右側の11ビットには、符号が格納されていない。また、矢印A19により示される2つのベクトル領域のうちの下側のベクトル領域の左側の8ビットの領域、左から2番目の8ビットの領域、右から2番目の8ビットの領域、および一番右側の8ビットの領域のそれぞれの右側の6ビットの領域にも、符号が格納されていない。

30

【0274】

このように、スカラ領域およびベクトル領域に格納されて記憶されている符号を、符号化された画像として出力する場合に、入力された量子化係数の符号化が終了した時点において、スカラ領域およびベクトル領域に符号が格納されていない領域があるときには、その符号が格納されていない領域に、例えば符号“0”などの任意の符号が格納されて記憶されてから、スカラ領域およびベクトル領域に記憶された符号が符号化された画像として出力される。

40

【0275】

したがって、例えば、矢印A19に示す例においては、符号化された画像として、スカラ領域に格納された符号“11000110100001101010100000000000”、図中、上側のベクトル領域に記憶された符号“10101111011110011100000101001110”、および下側のベクトル領域に記憶された符号“01000000000000011000000100000000”が順番に出力される。ここで、スカラ領域およびベクトル領域のうち、量子化係数の符号化が終了した時点で符号が格納されていなかった領域に格納される任意の符号は、復号時に読み込まれることはないので、どのような符号が格納されてもよい。

【0276】

50

画像符号化装置11は、SIMD演算命令を利用して量子化係数の絶対値を符号化する場合においても、画像が入力されると、図5のフローチャートを参照して説明した符号化処理を行う。また、図5のステップS13の処理に対応する図6のエントロピ符号化処理においては、図6のステップS41乃至ステップS49の各処理、ステップS51の処理、およびステップS52の処理については、画像符号化装置11は、SIMD演算命令を利用しない場合と同様の処理（図6を参照して説明した処理）を行い、ステップS50に対応するw個組符号化処理において、SIMD演算命令を利用しない場合とは異なる処理を行う。

【0277】

以下、図17のフローチャートを参照して、画像符号化装置11がSIMD演算命令を利用して量子化係数の絶対値を符号化する場合における、w個組符号化処理について説明する。なお、ステップS311乃至ステップS319の各処理は、図7のステップS81乃至ステップS89の各処理にそれぞれ対応し、それぞれ同様に実行される。従って、それらの説明は繰り返しになるので省略する。

10

【0278】

また、SIMD演算命令を利用して量子化係数の絶対値を符号化する場合、図6または図17を参照して説明した処理において、VLC符号化部62から符号連結部69に供給される、これから符号化するラインの量子化係数の絶対値が全て0であるか否かを示す符号、VLC符号化部64から符号連結部69に供給される量子化係数の最大有効桁数を示す符号、およびVLC符号化部68から符号連結部69に供給される量子化係数のサインを示す符号のそれぞれは、図16を参照して説明したように、符号連結部69のバッファ301に設けられたスカラ領域に格納されて記憶される。

20

【0279】

ステップS320において、有効桁抽出部65は、量子化部22から供給された、w個の連続するラインy上の位置 $(x, y)$ 、 $(x+1, y)$ 、 $\dots$ 、 $(x+w-1, y)$ の量子化係数から、量子化係数の有効桁を同時に抽出する。有効桁抽出部65は、抽出した量子化係数の有効桁をVLC符号化部66およびサイン抽出部67に供給する。また、VLC符号化部66は、有効桁抽出部65から供給された有効桁に基づいて（有効桁を符号化して）、w個の量子化係数の絶対値を示す符号を同時に符号連結部69に出力する。

【0280】

ここで、位置 $(x, y)$ におけるxの値は、最大有効桁数計算部63が記憶している変数xの値とされ、yの値は、ライン判定部61が記憶している変数yの値とされる。例えば、有効桁抽出部65が量子化係数の有効桁として、有効桁“101”、“011”、“110”、および“010”を抽出した場合、VLC符号化部66から符号連結部69には4つの量子化係数の絶対値を示す符号“101”、“011”、“110”、および“010”が供給されるので、符号連結部69は、供給されたこれらの量子化係数の絶対値を示す符号を符号化して、図16の矢印A12に示したように、ベクトル領域に格納して記憶させる。

30

【0281】

ステップS321において、サイン抽出部67は、所定の変数をiとして、変数iを0から $(w-1)$ まで変化させて、量子化部22から供給された、量子化係数が0でないラインy上の位置 $(x+i, y)$ の量子化係数から、量子化係数のサインを抽出し、抽出したサイン（のデータ）をVLC符号化部68に供給する。ここで、位置 $(x, y)$ におけるxの値は、最大有効桁数計算部63が記憶している変数xの値とされ、yの値は、ライン判定部61が記憶している変数yの値とされる。

40

【0282】

VLC符号化部68は、サイン抽出部67からのサインを符号化し、これにより得られる量子化係数のサインを示す符号を符号連結部69に出力する。また、符号連結部69は、図16を参照して説明したように、VLC符号化部68から供給された量子化係数のサインを示す符号を、バッファ301のスカラ領域に格納して記憶させる。

【0283】

符号連結部69は、量子化係数のサインを示す符号をバッファ301のスカラ領域に格納し

50

て記憶させると、図16を参照して説明したように、バッファ301のスカラ領域に記憶されている符号、およびベクトル領域に記憶されている符号を連結して、連結された符号を符号化された画像として出力し、w個組符号化処理を終了し、処理を図6のステップS50に戻し、ステップS51以降の処理を実行させる。

【0284】

このようにして、エントロピ符号化部23は、予め定められた数の量子化係数の絶対値を同時に符号化する。

【0285】

従来のJPEG2000方式によるエントロピ符号化においては、複数のコーディングパスに基づいて、量子化係数をビットプレーンごとに算術符号化するため、エントロピ符号化における所定の処理を並列して同時に行うことは困難であったが、エントロピ符号化部23においては、ビットプレーン単位で複雑な処理を行う必要はないので、複数の量子化係数の絶対値を同時に符号化することができる。

【0286】

このように、予め定められた数の量子化係数の絶対値を同時に符号化することによって、複数の処理を同時（並列）に行うことができ、より高速に画像を符号化することができる。

【0287】

なお、ステップS321の処理において、w個の量子化係数のサインの符号化を順番に行うと説明したが、量子化係数の絶対値の符号化における場合と同様に、SIMD演算命令を利用することで、w個の量子化係数のサインの符号化を同時に行うようにしてもよい。この場合、符号化により得られたw個の量子化係数のサインを示す符号のそれぞれは、バッファ301のベクトル領域にw個に分けられて格納される。

【0288】

また、バッファ301においては、1つのスカラ領域またはベクトル領域を32ビットの領域とし、その32ビットの領域をさらに4つの8ビットの領域に分けて使用すると説明したが、1つのスカラ領域またはベクトル領域の大きさなどは任意の大きさとすることができる。例えば、1つのスカラ領域またはベクトル領域を128ビットの領域とし、その128ビットの領域を8つの16ビットの領域に分けて使用するようにしてもよい。

【0289】

さらに、SIMD演算命令を利用して符号化された画像を復号する場合、画像を復号する画像復号装置111の符号分割部151（図9）は、例えば、図10に示した構成とされ、メモリ172には、符号化された画像としての符号が、図16を参照して説明したように、32ビットずつ格納されて記憶される。

【0290】

制御部171は、メモリ172から符号を読み出して出力する場合、まず、最初の32ビットの符号が格納されている記憶領域をスカラ領域として、スカラ領域の先頭から順番に、これから復号するラインの量子化係数の絶対値が全て0であるか否かを示す符号、量子化係数の最大有効桁数を示す符号、または量子化係数のサインを示す符号を読み出して出力する。

【0291】

また、制御部171は、メモリ172から量子化係数の絶対値を示す符号を読み出す場合、メモリ172のスカラ領域とされた記憶領域の次の32ビットの記憶領域（したがって、この記憶領域からはまだ符号が読み出されていない）をベクトル領域として、このベクトル領域から量子化係数の絶対値を示す符号を読み出して出力する。

【0292】

なお、画像が符号化される場合、復号時において最初に量子化係数の絶対値を示す符号を読み出すときに、スカラ領域とされた記憶領域の次の32ビットの記憶領域に、必ず量子化係数の絶対値を示す符号（ベクトル演算に用いられる符号）が格納されているように画像の符号化が行われる。

10

20

30

40

50

## 【0293】

さらに、符号化された画像としての符号が何ビットの記憶領域に分けられてメモリ172に格納されるかは、画像符号化装置11において画像が符号化される場合に、1つのスカラー領域およびベクトル領域が何ビットとされるかに応じて変化する。すなわち、画像としての符号が分けられて格納されるメモリ172内の複数の記憶領域のそれぞれの大きさは、画像が符号化される場合における1つのスカラー領域およびベクトル領域の大きさと同じ大きさとされる。

## 【0294】

画像復号装置111は、SIMD演算命令を利用して量子化係数の絶対値を復号する場合においても、符号化された画像が入力されると、図12のフローチャートを参照して説明した復号処理を行う。また、図12のステップS131の処理に対応する図13のエントロピ復号処理においては、図13のステップS161乃至ステップS168の各処理、ステップS170の処理、並びにステップS171の処理については、画像復号装置111は、SIMD演算命令を利用しない場合と同様の処理（図13を参照して説明した処理）を行い、ステップS169に対応するw個組復号処理において、SIMD演算命令を利用しない場合とは異なる処理を行う。

10

## 【0295】

以下、図18のフローチャートを参照して、画像復号装置111がSIMD演算命令を利用して量子化係数の絶対値を復号する場合における、w個組復号処理について説明する。

## 【0296】

なお、ステップS351乃至ステップS355の各処理は、図14のステップS211乃至ステップS215の各処理とそれぞれ対応し、それぞれ同様に実行される。従って、それらについての説明は繰り返しになるので省略する。

20

## 【0297】

また、SIMD演算命令を利用して量子化係数の絶対値を復号する場合、符号分割部151のメモリ172には、例えば、図16の矢印A19に示したように、画像としての符号が3つの32ビットの領域に分けられて記憶される。そして、ライン判定部152、VLC復号部154、およびVLC復号部156のそれぞれは、3つの32ビットの領域のうち、図16中、一番上の領域をスカラー領域として、スカラー領域の先頭（図中、左側）からラインの量子化係数が全て0であるか否かを示す符号、量子化係数の最大有効桁数を示す符号、および量子化係数のサインを示す符号のそれぞれを順番に読み出して復号する。

30

## 【0298】

ステップS356において、VLC復号部155は、符号分割部151から、連続するBビットの符号をw個同時に読み込み、読み込んだw個のBビットの符号のそれぞれを、ラインy上の位置（ $x, y$ ）、（ $x+1, y$ ）、 $\dots$ 、（ $x+w-1, y$ ）の量子化係数の有効桁を示す符号として、VLC復号部156および量子化係数合成部157に供給（出力）する。また、VLC復号部155は、量子化係数の有効桁を示す情報を生成して、符号分割部151に供給する。ここで、位置（ $x, y$ ）におけるxの値は、VLC復号部154が記憶している変数xの値とされ、yの値は、ライン判定部152が記憶している変数yの値とされる。

## 【0299】

例えば、予め定められた数wが4であり、変数Bの値が3であり、符号分割部151のメモリ172に、図16の矢印A19に示したように、画像としての符号が3つの32ビットの記憶領域に分けられて記憶されているとすると、図16中、一番上側の32ビットの記憶領域は、既にスカラー領域とされて、ラインの量子化係数が全て0であるか否かを示す符号、および量子化係数の最大有効桁数を示す符号が読み出されており、次の32ビットの記憶領域（上から2番目の記憶領域）からは、まだ符号が読み出されていないので、VLC復号部155は、上から2番目の記憶領域をベクトル領域として、このベクトル領域の左側の8ビットの領域、左から2番目の8ビットの領域、右から2番目の8ビットの領域、および一番右側の8ビットの領域のそれぞれの図中左側から、位置（ $x, y$ ）、（ $x+1, y$ ）、（ $x+2, y$ ）、および（ $x+3, y$ ）の量子化係数の有効桁を示す符号“101”、“011”、“110”、および“010”のそれぞれを同時に読み出して出力する。

40

50

## 【0300】

w個の量子化係数の有効桁を示す符号が、VLC復号部156および量子化係数合成部157に供給されると、その後、ステップS357の処理およびステップS358の処理が行われるが、これらの処理は、図14のステップS217の処理およびステップS218の処理と同様なので、その説明は省略する。

## 【0301】

このようにして、エントロピ復号部121は、予め定められた数の量子化係数の絶対値を同時に復号する。

## 【0302】

このように、予め定められた数の量子化係数の絶対値を同時に復号することによって、複数の処理を同時（並列）に行うことができ、より高速に画像を復号することができる。

10

## 【0303】

なお、ステップS357の処理において、w個の量子化係数のサインを示す符号のそれぞれの復号を順番に行うと説明したが、w個の量子化係数のサインを示す符号のそれぞれの復号を、SIMD演算命令を利用して同時に（並列に）行うようにしてもよい。

## 【0304】

以上のように、従来のJPEG2000方式による画像の符号化（または復号）とは異なり、複数のコーディングパスに基づいて、量子化係数をビットプレーンごとに算術符号化する必要がないため、より簡単な処理でより高速に画像を符号化（または復号）することができる。

20

## 【0305】

従来のJPEG2000方式においては、複数のコーディングパスに基づいて、ビットプレーン毎に処理を行うので、それらの処理を行う場合に、おおよそ量子化係数にビットプレーンの数を掛け合わせた回数程度、量子化係数にアクセスしなければならず、処理量が多かった。

## 【0306】

また、符号化された画像をパケット化する場合に、1つの画像の符号化が完全に終了しなければパケット化の処理を開始することができないため、その分だけ遅延が生じる。さらに、JPEG2000方式においては、例えば、図2に示したサブバンド上のx方向およびy方向に平行な辺からなる矩形領域内の位置に対応する（符号化された）量子化係数が1つのパケットに格納されるため、その矩形領域のy方向の長さに応じた遅延も生じる。従来のJPEG2000方式においては、このような符号化による遅延が生じるため、リアルタイムでの処理は困難であった。なお、サブバンド上の矩形領域のy方向の長さを短くすることにより、遅延を低減することは可能であったが、この場合、符号化効率が悪くなってしまう。

30

## 【0307】

これに対して、画像符号化装置11においては、上述したように、複数のコーディングパスに基づいて、量子化係数をビットプレーンごとに算術符号化する必要はなく、画像の符号化を行う場合に、量子化係数の絶対値を示す符号を出力するとき、最大有効桁数を示す符号を出力するとき、および量子化係数のサインを示す符号を出力するときに量子化係数にアクセスするだけなので、より簡単に画像を符号化することができる。

40

## 【0308】

なお、最大有効桁数を示す符号および量子化係数のサインを示す符号は1ビットや0ビットであることもあるので、実質的には、画像の符号化を行う場合に、おおよそ2回程度量子化係数にアクセスするだけで画像を符号化することができる。また、画像を復号する場合、量子化係数には1度だけアクセスすればよいので、より簡単により高速に画像を復号することができる。

## 【0309】

画像符号化装置11または画像復号装置111においては、サブバンドの量子化係数をラスタスキャン順に符号化または復号するので、量子化係数をバッファリングする必要がなく、符号化または復号による遅延を低減することができる。

50



## 【0310】

さらに、実際に横1920画素×縦1080画素のYUV4:2:2形式の画像を、SIMD演算命令を利用して（但し、 $w=4$ ）実際に符号化および復号したところ、次のような結果が得られた。なお、符号化を行う場合、画像をウェーブレット変換して5段階のサブバンド分解を行い、さらに、各サブバンドのウェーブレット係数を量子化して得られた量子化係数を符号化した。また、符号化および復号に必要な機能ブロック（例えば、図15のエントロピ符号化部23および図9のエントロピ復号部121）、並びにJPEG2000方式により画像を符号化および復号する機能ブロックは、Pentium（登録商標）4（インテル株式会社の商標）と称されるCPU（クロック周波数3.0GHz）に所定のプログラムを実行させることにより実現させた。

## 【0311】

従来のJPEG2000方式により1フレーム分の画像を符号化した場合、その符号量は291571バイトであり、符号化に必要とされた時間は、0.26157秒であった。また、符号化された画像を復号するために必要な時間は0.24718秒であった。

## 【0312】

これに対して、図15のエントロピ符号化部23により、1フレーム分の画像を符号化した場合、その符号量は343840バイトであり、符号化に必要とされた時間は、0.03453秒であった。また、符号化された画像を、図9のエントロピ復号部121により復号するために必要な時間は0.02750秒であった。

## 【0313】

動画像においては、多くの場合、1秒間に30フレーム表示されるので、1フレーム当たり0.033（ $=1/30$ ）秒で符号化または復号することができれば、画像をリアルタイムに処理することができる。JPEG2000方式では、符号化に必要な時間は0.26157秒であり、復号に必要な時間は0.24718秒であるので、画像をリアルタイムに処理することは困難であるが、図15のエントロピ符号化部23により画像を符号化させる場合には、符号化に必要な時間は0.03453秒であり、ほぼリアルタイムに画像を処理することができる。また、図9のエントロピ復号部121により画像を復号する場合、復号に必要な時間は0.02750秒であるので、十分にリアルタイムで処理を行うことができる。

## 【0314】

なお、以上においては、量子化係数を予め定められた所定の数ずつまとめて符号化、または復号する処理をJPEG2000方式に適用する例について説明したが、JPEG2000方式に限らず、他の符号化方式または復号方式にも適用することが可能である。

## 【0315】

また、以上においては、画像データを符号化または復号する例について説明したが、画像データに限らず、例えば、音声データなどを符号化または復号する場合にも適用することが可能である。例えば、音声データを符号化する場合、音声データとして入力された符号により表わされる $w$ 個の所定の数値の最大有効桁数を示す符号、それらの数値の絶対値を示す符号、およびそれらの数値のサインを示す符号が符号化された音声データとして出力される。

## 【0316】

ここで、本発明の符号化処理の特徴についてさらに付言すると、本発明の符号化方式では、量子化係数が可逆符号化される。そのため、例えば、人間の視覚特性に合わせて高域の係数ほど大きな量子化ステップサイズで量子化されるようにすることにより、発生符号量あたりの画質を大きく向上させることが出来る。また、例えば、特定の空間範囲で用いる量子化ステップサイズを小さくすることにより、その空間範囲の画質を向上させることもできる。

## 【0317】

さらに、本実施の形態において説明した符号化方式では、絶対値の有効桁部分を並べたものが符号される。仮に、絶対値の有効桁部分がVLC符号化されて送出されるようにすると、絶対値の有効桁数が $N$ の場合には、 $2^{(N \times W)}$ 個のエントリを持つ非常に大きなVLCテーブルが必要になる（演算処理の負荷や処理時間だけでなく、VLCテーブルを保持するのに

10

20

30

40

50

必要なメモリの容量も増大する)。これに対して、本発明の符号化方式においてはこのような大きなテーブルを使う必要がない(演算処理の負荷や処理時間だけでなく、メモリの容量を低減させることができる)。

【0318】

また、VLCよりも圧縮率が高い算術符号を使うことも考えられるが、例えばJPEG2000のような算術符号化を用いた圧縮方式を用いた場合でも、本発明の符号化方式の場合と比べて、圧縮率が大きく向上することはない。つまり、本発明の符号化方式は、符号化処理が容易でありながら圧縮率が高い。

【0319】

さらに、本発明の符号化方式は、w個組の係数における絶対値の最大有効桁数を符号化するので、隣同士の係数の有効桁数が似ていることを利用して発生符号量を低減させることができる。

10

【0320】

また、本発明の符号化方式は、w個組の係数における絶対値の最大有効桁数を符号化するとき差分符号化を行うので、この点でも隣同士の係数の有効桁数が似ていることを利用して発生符号量を低減させることができる。

【0321】

次に、画像符号化装置11のウェーブレット変換部21や、画像復号装置111のウェーブレット逆変換部123の具体的な例について説明する。このウェーブレット変換部21として、例えば、画面(ピクチャまたはフィールド)全体の画像データに対して画面垂直方向および画面水平方向に分析フィルタ処理を行い、画面全体を複数のサブバンドに分解する、一般的なウェーブレット変換処理を行うものを用いるようにしてもよいが、以下に説明するような方法でウェーブレット変換処理を行うものを用いる方がより高速に画像を符号化することができる。同様に、ウェーブレット逆変換部123として、例えば、画面(ピクチャまたはフィールド)全体の係数データに対して画面垂直方向および画面水平方向に合成フィルタ処理を行い、各サブバンドを合成する、一般的なウェーブレット逆変換処理を行うものを用いるようにしてもよいが、以下に説明するような方法でウェーブレット逆変換処理を行うものを用いる方がより高速に画像を復号することができる。

20

【0322】

図19は、本発明を適用した画像符号化装置の一例の構成を示すブロック図である。画像符号化装置401は、ウェーブレット変換部410、途中計算用バッファ部411、係数並び替え用バッファ部412、係数並び替え部413、レート制御部414、およびエントロピ符号化部415からなる。

30

【0323】

つまり、図19の画像符号化装置401の、ウェーブレット変換部410、途中計算用バッファ部411、係数並び替え用バッファ部412、および係数並び替え部413は、図1の画像符号化装置11のウェーブレット変換部21に対応し、図19の画像符号化装置401のレート制御部414、およびエントロピ符号化部415は、図1の画像符号化装置11の量子化部22およびエントロピ符号化部23に対応する。

【0324】

40

入力された画像データは、途中計算用バッファ部411に一時的に溜め込まれる。ウェーブレット変換部410は、途中計算用バッファ部411に溜め込まれた画像データに対してウェーブレット変換を施す。すなわち、ウェーブレット変換部410は、途中計算用バッファ部411から画像データを読み出して分析フィルタによりフィルタ処理を施して低域成分および高域成分の係数のデータを生成し、生成された係数データを途中計算用バッファ部411に格納する。ウェーブレット変換部410は、水平分析フィルタと垂直分析フィルタとを有し、画像データ群に対して、画面水平方向と画面垂直方向の両方について分析フィルタ処理を行う。ウェーブレット変換部410は、途中計算用バッファ部411に格納された低域成分の係数データを再度読み出し、読み出した係数データに対して分析フィルタによるフィルタ処理を施して、高域成分および低域成分の係数のデータをさらに生成する。生成された係

50

数データは、途中計算用バッファ部411に格納される。

【0325】

ウェーブレット変換部410は、この処理を繰り返して分解レベルが所定レベルに達したら、途中計算用バッファ部411から係数データを読み出し、読み出された係数データを係数並び替え用バッファ部412に書き込む。

【0326】

係数並び替え部413は、係数並び替え用バッファ部412に書き込まれた係数データを所定の順序で読み出し、エントロピ符号化部415に供給する。エントロピ符号化部415は、供給された係数データを、例えばハフマン符号化や算術符号化といった所定のエントロピ符号化方式で符号化する。

10

【0327】

エントロピ符号化部415は、レート制御部414と連動的に動作し、出力される圧縮符号化データのビットレートが略一定値となるように制御される。すなわち、レート制御部414は、エントロピ符号化部415からの符号化データ情報に基づき、エントロピ符号化部415により圧縮符号化されたデータのビットレートが目標値に達した時点あるいは目標値に達する直前でエントロピ符号化部415による符号化処理を終了するように制御する制御信号を、エントロピ符号化部415に対して供給する。エントロピ符号化部415は、レート制御部414から供給される制御信号に応じて符号化処理が終了した時点で、符号化データを出力する。

【0328】

20

ウェーブレット変換部410で行われる処理について、より詳細に説明する。まず、ウェーブレット変換について、概略的に説明する。画像データに対するウェーブレット変換では、図20に概略的に示されるように、画像データを空間周波数の高い帯域と低い帯域とに分割する処理を、分割の結果得られる空間周波数の低い帯域のデータに対して再帰的に繰り返す。こうして、空間周波数の低い帯域のデータをより小さな領域に追い込んでいくことで、効率的な圧縮符号化を可能とする。

【0329】

なお、図20は、画像データの最低域成分領域に対する低域成分の領域Lおよび高域成分の領域Hへの分割処理を3回、繰り返し、分割レベル=3とした場合の例である。図20において、"L"および"H"は、それぞれ低域成分および高域成分を表し、"L"および"H"の順序は、前側が横方向に分割した結果の帯域を示し、後側が縦方向に分割した結果の帯域を示す。また、"L"および"H"の前の数字は、その領域の分割レベルを示す。

30

【0330】

また、図20の例から分かるように、画面の右下の領域から左上の領域にかけて段階的に処理がなされ、低域成分が追い込まれていく。すなわち、図20の例では、画面の右下の領域が最も低域成分の少ない（高域成分が最も多く含まれる）領域3HHとされる、画面が4分割された左上の領域は、さらに4分割され、この4分割された領域のうち左上の領域がさらに4分割される。最も左上隅の領域は、最も低域成分を多く含む領域0LLとされる。

【0331】

低域成分に対して繰り返し変換および分割を行うのは、画像のエネルギーが低域成分に集中しているためである。このことは、図21のAに一例が示される分割レベル=1の状態から、図21のBに一例が示される分割レベル=3の状態のように分割レベルを進めていくに従って、図21のBに示されるようにしてサブバンドが形成されていくことから、理解される。例えば、図20におけるウェーブレット変換の分割レベルは3であり、この結果、10個のサブバンドが形成されている。

40

【0332】

ウェーブレット変換部410は、通常、低域フィルタと高域フィルタとから構成されるフィルタバンクを用いて、上述のような処理を行う。なお、デジタルフィルタは、通常、複数次のインパルス応答すなわちフィルタ係数を持っているため、フィルタ処理を行えるだけの入力画像データまたは係数データを予めバッファリングしておく必要がある

50

。また、ウェーブレット変換を多段にわたって行う場合も同様に、前段で生成したウェーブレット変換係数を、フィルタ処理が行える数だけバッファリングしておく必要がある。

【 0 3 3 3 】

次に、このウェーブレット変換の具体的な例として、 $5 \times 3$ フィルタを用いた方法について説明する。この $5 \times 3$ フィルタを用いた方法は、従来技術で既に説明したJPEG2000規格でも採用されており、少ないフィルタタップ数でウェーブレット変換を行うことができる点で、優れた方法である。

【 0 3 3 4 】

$5 \times 3$ フィルタのインパルス応答（Z変換表現）は、次の式（3）および式（4）に示すように、低域フィルタ $H_0(z)$ と、高域フィルタ $H_1(z)$ とから構成される。式（3）および式（4）から、低域フィルタ $H_0(z)$ は、5タップで、高域フィルタ $H_1(z)$ は、3タップであることが分かる。

【 0 3 3 5 】

$$H_0(z) = (-1 + 2z^{-1} + 6z^{-2} + 2z^{-3} - z^{-4}) / 8 \quad \cdots (3)$$

$$H_1(z) = (-1 + 2z^{-1} - z^{-2}) / 2 \quad \cdots (4)$$

【 0 3 3 6 】

これら式（3）および式（4）によれば、低域成分および高域成分の係数を、直接的に算出することができる。ここで、リフティング(Lifting)技術を用いることで、フィルタ処理の計算を減らすことができる。図22を用いて、 $5 \times 3$ フィルタに対してリフティング技術を適用した場合の、ウェーブレット変換を行う分析フィルタ側の処理について、概略的に説明する。

【 0 3 3 7 】

図22において、最上段部、中段部および最下段部は、それぞれ入力画像の画素列、高域成分出力および低域成分出力を示す。最上段は、入力画像の画素列に限らず、先のフィルタ処理で得られた係数でもよい。ここでは、最上段部が入力画像で画素列であるものとし、四角印（ ）が偶数番目（最初を0番目とする）の画素またはライン、丸印（ ）が奇数番目の画素またはラインとする。

【 0 3 3 8 】

まず第1段階として、次式（5）により入力画素列から高域成分の係数 $d_i^1$ を生成する。

$$d_i^1 = d_i^0 - 1/2(s_i^0 + s_{i+1}^0) \quad \cdots (5)$$

【 0 3 3 9 】

次に第2段階として、この生成された高域成分の係数と、入力画像の奇数番目の画素とを用いて、次式（6）により低域成分の係数 $s_i^1$ を生成する。

$$s_i^1 = s_i^0 + 1/4(d_{i-1}^1 + d_i^1) \quad \cdots (6)$$

【 0 3 4 0 】

分析フィルタ側では、このようにして、フィルタリング処理により入力画像の画素データを低域成分と高域成分とに分解する。

【 0 3 4 1 】

図23を用いて、ウェーブレット変換により生成された係数を復元するウェーブレット逆変換を行う合成フィルタ側の処理について、概略的に説明する。この図23は、上述の図22と対応し、 $5 \times 3$ フィルタを用い、リフティング技術を適用した例を示す。図23において、最上段部は、ウェーブレット変換により生成された入力係数を示し、丸印（ ）が高域成分の係数、四角印（ ）が低域成分の係数をそれぞれ示す。

【 0 3 4 2 】

まず第1段階として、次式（7）に従い、入力された低域成分および高域成分の係数から、偶数番目（最初を0番目とする）の係数 $s_i^0$ が生成される。

$$s_i^0 = s_i^1 - 1/4(d_{i-1}^1 + d_i^1) \quad \cdots (7)$$

【 0 3 4 3 】

次に第2段階として、次式（8）に従い、上述の第1段階で生成された偶数番目の係数 $s_i^0$ と、入力された高域成分の係数 $d_i^1$ とから、奇数番目の係数 $d_i^0$ が生成される。

10

20

30

40

50

$$d_i^0 = d_i^1 + 1/2(s_i^0 + s_{i+1}^0) \quad \cdots (8)$$

【 0 3 4 4 】

合成フィルタ側では、このようにして、フィルタリング処理により低域成分および高域成分の係数を合成し、ウェーブレット逆変換を行う。

【 0 3 4 5 】

さらに、このウェーブレット変換方法について説明する。図24は、図22を用いて説明した5×3フィルタのリフティングによるフィルタ処理を、分解レベル=2まで実行した例を示す。なお、図24において、図の左側に分析フィルタとして示される部分は、画像符号化装置401側におけるウェーブレット変換部410のフィルタである。また、図の右側に合成フィルタとして示される部分は、後述する画像復号装置側におけるウェーブレット逆変換部のフィルタである。

【 0 3 4 6 】

なお、以下の説明では、例えば表示デバイスなどにおいて画面の左上隅の画素を先頭として、画素が画面の左端から右端に向けて走査されて1ラインが構成され、ライン毎の走査が画面の上端から下端に向けて行われて1画面が構成されるものとする。

【 0 3 4 7 】

図24において、左端列は、原画像データのライン上の対応する位置にある画素データが縦方向に並べられて示されている。すなわち、ウェーブレット変換部410におけるフィルタ処理は、垂直フィルタを用いて画面上を画素が縦に走査されて行われる。左端から1列目乃至3列目が分割レベル=1のフィルタ処理を示し、4列目乃至6列目が分割レベル=2のフィルタ処理を示す。左端から2列目は、左端の原画像データの画素に基づく高域成分出力、左端から3列目は、原画像データおよび高域成分出力に基づく低域成分出力を示す。分割レベル=2のフィルタ処理は、左端から4列目乃至6列目に示されるように、分割レベル=1のフィルタ処理の出力に対して処理がなされる。

【 0 3 4 8 】

分解レベル=1のフィルタ処理において、第1段階のフィルタ処理として、原画像データの画素に基づき高域成分の係数データが算出され、第2段階のフィルタ処理として、第1段階のフィルタ処理で算出された高域成分の係数データと、原画像データの画素とに基づき低域成分の係数データが算出される。分解レベル=1の一例のフィルタ処理を、図24における左側（分析フィルタ側）の第1列目乃至第3列目に示す。算出された高域成分の係数データは、図19で説明した係数並び替え用バッファ部412に格納される。また、算出された低域成分の係数データのは、途中計算用バッファ部411に格納される。

【 0 3 4 9 】

図24においては、係数並び替え用バッファ部412は、一点鎖線で囲まれた部分として示し、途中計算用バッファ部411は、点線で囲まれた部分として示す。

【 0 3 5 0 】

途中計算用バッファ部411に保持された分解レベル=1のフィルタ処理の結果に基づき、分解レベル=2のフィルタ処理が行われる。分解レベル=2のフィルタ処理では、分解レベル=1のフィルタ処理において低域成分の係数として算出された係数データを、低域成分および高域成分を含んだ係数データと見做して、分解レベル=1と同様のフィルタ処理を行う。分解レベル=2のフィルタ処理により算出された、高域成分の係数データおよび低域成分の係数データは、図19で説明した係数並び替え用バッファ部412に格納される。

【 0 3 5 1 】

ウェーブレット変換部410では、上述したようなフィルタ処理を、画面の水平方向および垂直方向にそれぞれ行う。例えば、まず、分解レベル=1のフィルタ処理を水平方向に行い、生成された高域成分および低域成分の係数データを途中計算用バッファ部411に格納する。次に、途中計算用バッファ部411に格納された係数データに対して、垂直方向に分解レベル=1のフィルタ処理を行う。この分解レベル=1の水平および垂直方向の処理により、高域成分をさらに高域成分および低域成分に分解した係数データのそれぞれによる領域HHおよび領域HLと、低域成分をさらに高域成分および低域成分に分解した係数データ

のそれぞれによる領域LHおよび領域LLとの4領域が形成される。

【0352】

そして、分解レベル=2では、水平方向および垂直方向のそれぞれについて、分解レベル=1で生成された低域成分の係数データに対してフィルタ処理が行われる。すなわち、分解レベル=2では、分解レベル=1で分割されて形成された領域LLがさらに4分割され、領域LL内にさらに領域HH、領域HL、領域LHおよび領域LLが形成される。

【0353】

この実施の第1の形態では、ウェーブレット変換によるフィルタ処理を、画面の縦方向について、数ライン毎の処理に分割して、複数回に分けて段階的に行うようにしている。図24の例では、画面上の第1ラインからの処理になる1回目の処理は、7ラインについてフ  
10  
ィルタ処理を行い、8ライン目からの処理になる2回目以降の処理は、4ライン毎にフ  
ィルタ処理を行っている。このライン数は、高域成分と低域成分とに2分解した後、1ライン分の最低域成分が生成されるために必要なライン数に基づく。

【0354】

なお、以下において、この最低域成分の1ライン分（最低域成分のサブバンドの1ライン分の係数データ）を生成するために必要な、他のサブバンドも含めたラインの集まりを、ラインブロック（またはプレシント）と称する。ここでラインとは、ウェーブレット変換前の画像データに対応するピクチャ若しくはフィールド内、または各サブバンド内において形成される1行分の画素データ若しくは係数データのことを示す。すなわち、ライン  
20  
ブロック（プレシント）とは、ウェーブレット変換前の元の画像データにおける、ウェ  
ーブレット変換後の最低域成分のサブバンド1ライン分の係数データを生成するために必要なライン数分の画素データ群、または、その画素データ群をウェーブレット変換して得られる各サブバンドの係数データ群のことを示す。

【0355】

図24によれば、分解レベル=2のフィルタ処理結果で得られる係数C5は、係数C4および途中計算用バッファ部411に格納された係数C<sub>a</sub>に基づき算出され、係数C4は、途中計算用バッファ部411に格納された係数C<sub>a</sub>、係数C<sub>b</sub>および係数C<sub>c</sub>に基づき算出される。さらに、係数C<sub>c</sub>は、係数並び替え用バッファ部412に格納される係数C2および係数C3、並びに、第5ラインの画素データに基づき算出される。また、係数C3は、第5ライン乃至第7ラインの画  
30  
素データに基づき算出される。このように、分割レベル=2における低域成分の係数C5を得るためには、第1ライン乃至第7ラインの画素データが必要とされる。

【0356】

一方、2回目以降のフィルタ処理においては、前回までのフィルタ処理で既に算出され係数並び替え用バッファ部412に格納されている係数データを用いることができるので、必要なライン数が少なくて済む。

【0357】

すなわち、図24によれば、分解レベル=2のフィルタ処理結果で得られる低域成分の係数のうち、係数C5の次の係数である係数C9は、係数C4および係数C8、並びに、途中計算用バッファ部411に格納された係数C<sub>c</sub>に基づき算出される。係数C4は、上述した1回目のフ  
40  
ィルタ処理により既に算出され、係数並び替え用バッファ部412に格納されている。同様に、係数C<sub>c</sub>は、上述の1回目のフィルタ処理により既に算出され、途中計算用バッファ部411に格納されている。したがって、この2回目のフィルタ処理においては、係数C8を算出するためのフィルタ処理のみが、新たになされることになる。この新たなフィルタ処理は、第8ライン乃至第11ラインがさらに用いられてなされる。

【0358】

このように、2回目以降のフィルタ処理は、前回までのフィルタ処理により算出され途中計算用バッファ部411および係数並び替え用バッファ部412に格納されたデータを用いることができるので、それぞれ4ライン毎の処理で済むことになる。

【0359】

なお、画面上のライン数が符号化のライン数と合致しない場合は、原画像データのライ  
50

ンを所定に複製してライン数を符号化のライン数と合わせて、フィルタ処理を行う。

【0360】

詳細は後述するが、この発明では、このように、最低域成分1ライン分の係数データが得られるだけのフィルタ処理を段階的に、画面全体のラインに対して複数回に分けて（ラインブロック単位で）行うことで、符号化データを伝送した際に低遅延で復号画像を得ることを可能としている。

【0361】

ウェーブレット変換を行うためには、ウェーブレット変換そのものを実行するために用いられる第1のバッファと、所定の分割レベルまで処理を実行する間に生成される係数を格納するための第2のバッファとが必要とされる。第1のバッファは、途中計算用バッファ部411に対応し、図24においては点線で囲まれて示されている。また、第2のバッファは、係数並び替え用バッファ部412に対応し、図24においては一点鎖線で囲まれて示されている。第2のバッファに格納された係数は、復号の際に用いられるため、後段のエントロピ符号化処理の対象とされる。

【0362】

係数並び替え部413の処理について説明する。上述したように、ウェーブレット変換部410で算出された係数データは、係数並び替え用バッファ部412に格納され、係数並び替え部413により順序を並び替えられて読み出され、エントロピ符号化部415に送出される。

【0363】

既に説明したように、ウェーブレット変換においては、高域成分側から低域成分側へと係数が生成されていく。図24の例では、1回目において、原画像の画素データにより、分解レベル=1のフィルタ処理で、高域成分の係数C1、係数C2および係数C3が順次生成される。そして、分解レベル=1のフィルタ処理で得られた低域成分の係数データに対して分解レベル=2のフィルタ処理を行い、低域成分の係数C4および係数C5が順次生成される。すなわち、第1回目では、係数C1、係数C2、係数C3、係数C4、係数C5の順に、係数データが生成される。この係数データの生成順は、ウェーブレット変換の原理上、必ずこの順序（高域から低域の順）になる。

【0364】

これに対して、復号側では、低遅延で即座に復号を行うためには低域成分から画像の生成および出力を行う必要がある。そのため、符号化側で生成された係数データを最低域成分側から高域成分側に向けて並び替えて復号側に供給することが望ましい。

【0365】

図24の例を用いて、より具体的に説明する。図24の右側は、逆ウェーブレット変換を行う合成フィルタ側を示す。復号側の、出力画像データの第1ライン目を含む1回目の合成処理（逆ウェーブレット変換処理）は、符号化側の1回目のフィルタ処理で生成された最低域成分の係数C4および係数C5と、係数C1とを用いて行われる。

【0366】

すなわち、1回目の合成処理においては、係数C5、係数C4、係数C1の順に符号化側から復号側に係数データを供給し、復号側では、分解レベル=2に対応する合成処理である合成レベル=2の処理で、係数C5および係数C4に対して合成処理を行って係数C<sub>f</sub>を生成し、バッファに格納する。そして、分解レベル=1に対応する合成処理である合成レベル=1の処理で、この係数C<sub>f</sub>と係数C1に対して合成処理を行って、第1ラインを出力する。

【0367】

このように、第1回目の合成処理においては、符号化側で係数C1、係数C2、係数C3、係数C4、係数C5の順に生成され係数並び替え用バッファ部412に格納された係数データが、係数C5、係数C4、係数C1、・・・の順に並び替えられて復号側に供給される。

【0368】

なお、図24の右側に示す合成フィルタ側では、符号化側から供給される係数について、括弧内に符号化側での係数の番号を記し、括弧外に合成フィルタのライン順を記す。例えば係数C1(5)は、図24の左側の分析フィルタ側では係数C5であって、合成フィルタ側で

10

20

30

40

50

は第1ライン目であることを示す。

【0369】

符号化側の2回目以降のフィルタ処理で生成された係数データによる復号側の合成処理は、前回の合成処理の際に合成あるいは符号化側から供給された係数データを用いて行うことができる。図24の例では、符号化側の2回目のフィルタ処理で生成された低域成分の係数C8および係数C9を用いて行う、復号側の2回目の合成処理は、符号化側の1回目のフィルタ処理で生成された係数C2および係数C3がさらに必要とされ、第2ライン乃至第5ラインが復号される。

【0370】

すなわち、2回目の合成処理においては、係数C9、係数C8、係数C2、係数C3の順に符号化側から復号側に係数データを供給する。復号側では、合成レベル=2の処理において、係数C8および係数C9と、1回目の合成処理の際に符号化側から供給された係数C4とを用いて係数C<sub>g</sub>を生成し、バッファに格納する。この係数C<sub>g</sub>と、上述の係数C4と、1回目の合成処理により生成されバッファに格納された係数C<sub>f</sub>とを用いて係数C<sub>h</sub>を生成し、バッファに格納する。

【0371】

そして、合成レベル=1の処理において、合成レベル=2の処理で生成されバッファに格納された係数C<sub>g</sub>および係数C<sub>h</sub>と、符号化側から供給された係数C2（合成フィルタでは係数C6（2）と示されている）および係数C3（合成フィルタでは係数C7（3）と示されている）とを用いて合成処理が行われ、第2ライン乃至第5ラインが復号される。

【0372】

このように、第2回目の合成処理においては、符号化側で係数C2、係数C3、（係数C4、係数C5）、係数C6、係数C7、係数C8、係数C9の順に生成された係数データが、係数C9、係数C8、係数C2、係数C3、・・・の順に並び替えられて復号側に供給される。

【0373】

3回目以降の合成処理においても、同様にして、並び替えバッファ部12に格納された係数データが所定に並び替えられて復号部に供給され、4ラインずつ、ラインが復号される。

【0374】

なお、符号化側において画面の下端のラインを含むフィルタ処理（以下、最後の回と呼ぶ）に対応する復号側の合成処理では、それまでの処理で生成されバッファに格納された係数データを全て出力することになるため、出力ライン数が多くなる。図24の例では、最後の回に8ラインが出力される。

【0375】

なお、係数並び替え部413による係数データの並び替え処理は、例えば、係数並び替え用バッファ部412に格納された係数データを読み出す際の読み出しアドレスを、所定の順序に設定することでなされる。

【0376】

図25を用いて、上述までの処理をより具体的に説明する。図25は、5×3フィルタを用いて、分解レベル=2までウェーブレット変換によるフィルタ処理を施した例である。ウェーブレット変換部410において、図25のAに一例が示されるように、入力画像データの第1ラインから第7ラインに対して1回目のフィルタ処理が水平および垂直方向にそれぞれ行われる（図25のAのIn-1）。

【0377】

1回目のフィルタ処理の分解レベル=1の処理において、係数C1、係数C2、および係数C3の3ライン分の係数データが生成され、図25のBに一例が示されるように、分解レベル=1で形成される領域HH、領域HLおよび領域LHのそれぞれに配置される（図25のBのWT-1）。

【0378】

また、分解レベル=1で形成される領域LLは、分解レベル=2による水平および垂直方向のフィルタ処理でさらに4分割される。分解レベル=2で生成される係数C5および係数C4は



、分解レベル＝1による領域LL内において、領域LLに係数C5による1ラインが配置され、領域HH、領域HLおよび領域LHのそれぞれに、係数C4による1ラインが配置される。

【0379】

ウェーブレット変換部410による2回目以降のフィルタ処理では、4ライン毎にフィルタ処理が行われ（図25のAのIn-2・・・）、分解レベル＝1で2ラインずつの係数データが生成され（図25のBのWT-2）、分解レベル＝2で1ラインずつの係数データが生成される。

【0380】

図24の2回目の例では、分解レベル＝1のフィルタ処理で係数C6および係数C7の2ライン分の係数データが生成され、図25のBに一例が示されるように、分解レベル1で形成される領域HH、領域HLおよび領域LHの、1回目のフィルタ処理で生成された係数データの次から配置される。同様に、分解レベル＝1による領域LL内において、分解レベル＝2のフィルタ処理で生成された1ライン分の係数C9が領域LLに配置され、1ライン分の係数C8が領域HH、領域HLおよび領域LHにそれぞれ配置される。

【0381】

図25のBのようにウェーブレット変換されたデータを復号した際には、図25のCに一例が示されるように、符号化側の第1ライン乃至第7ラインによる1回目のフィルタ処理に対して、復号側の1回目の合成処理による第1ラインが出力される（図25のCのOut-1）。以降、符号化側の2回目から最後の回の前までのフィルタ処理に対して、復号側で4ラインずつが出力される（図25のCのOut-2・・・）。そして、符号化側の最後の回のフィルタ処理に対して、復号側で8ラインが出力される。

【0382】

ウェーブレット変換部410で高域成分側から低域成分側へと生成された係数データは、係数並び替え用バッファ部412に順次格納される。係数並び替え部413は、上述した係数データの並び替えが可能となるまで係数並び替え用バッファ部412に係数データが蓄積されると、係数並び替え用バッファ部412から合成処理に必要な順に並び替えて係数データを読み出す。読み出された係数データは、エントロピ符号化部415に順次、供給される。

【0383】

エントロピ符号化部415は、供給された係数データに対して、レート制御部414から供給される制御信号に基づき出力データのビットレートが目標ビットレートになるように符号化動作を制御して、図1乃至図18を参照して説明したようにエントロピ符号化を施す。エントロピ符号化された符号化データは、復号側に供給される。

【0384】

なお、このようにエントロピ符号化部415は、供給された係数データに対して量子化を行わずに符号化処理を行うことも可能であるが、図1を参照して説明したように、量子化部22と同様に最初に、係数データに対して量子化を行い、得られた量子化係数に対して、エントロピ符号化部23と同様の情報源符号化処理を施すようにすれば、さらに圧縮効果の向上を期待することができる。この量子化の方法としてはどのようなものを用いても良く、例えば、一般的な手段、つまり、以下の式（9）に示されるような、係数データWを量子化ステップサイズ で除算する手法を用いれば良い。

【0385】

量子化係数 =  $W / \dots$  (9)

【0386】

図24および図25を用いて説明したように、この発明の実施の第1の形態では、ウェーブレット変換部410は、画像データの複数ライン毎（ラインブロック毎）にウェーブレット変換処理を行う。エントロピ符号化部415では符号化された符号化データは、このラインブロック毎に出力される。すなわち、上述の、5×3フィルタを用い、分解レベル＝2まで処理を行った場合には、1画面のデータの出力において、最初が1ライン、2回目以降最後の回の前までが4ラインずつ、最後の回が8ラインの出力が得られる。

【0387】

なお、係数並び替え部413で並び替えられた後の係数データをエントロピ符号化する場合

合、例えば図24で示した1回目のフィルタ処理では、最初の係数C5のラインをエントロピ符号化する際には、未だ過去のラインすなわち既に係数データが生成されたラインが存在していない。したがって、この場合には、この1ラインだけをエントロピ符号化する。これに対して、係数C1のラインを符号化する際には、係数C5および係数C4のラインが過去のラインとなっている。これら近接する複数ラインは、似たデータで構成されていることが考えられるので、これら複数ラインを纏めてエントロピ符号化することは、有効である。

【0388】

また、上述では、ウェーブレット変換部410において、 $5 \times 3$ フィルタを用いてウェーブレット変換によるフィルタ処理を行う例について説明したが、これはこの例に限られない。例えば、ウェーブレット変換部410では、例えば $9 \times 7$ フィルタといった、さらにタップ数の長いフィルタを用いることができる。この場合、フィルタのタップ数が長ければ、フィルタに蓄積されるライン数も多くなるので、画像データの入力から符号化データの出力までの遅延時間が長くなることになる。

【0389】

また、上述では、説明のためウェーブレット変換の分解レベルを分解レベル=2としたが、これはこの例に限られず、さらに分解レベルを上げることができる。分解レベルを上げるほど、より高圧縮率を実現することができる。例えば、一般的には、ウェーブレット変換においては、分解レベル=4までフィルタ処理が繰り返される。なお、分解レベルが上げれば、遅延時間も増大することになる。

【0390】

したがって、実際のシステムに適用する際には、当該システムに要求される遅延時間や復号画像の画質などに応じて、フィルタのタップ数や、分解レベルを決めることが好ましい。このフィルタのタップ数や、分解レベルは、固定値とせず、適応的に選択するようにもできる。

【0391】

次に、以上のような画像符号化装置401による符号化処理全体の具体的な流れの例を図26のフローチャートを参照して説明する。

【0392】

符号化処理が開始されると、ウェーブレット変換部410は、ステップS401において、処理対象ラインブロックの番号Aを初期設定にする。通常の場合、番号Aは「1」に設定される。設定が終了すると、ウェーブレット変換部410は、ステップS402において、最低域サブバンドにおいて上からA番目の1ラインを生成するのに必要なライン数（すなわち、1ラインブロック）の画像データを取得し、その画像データに対して、ステップS403において画面垂直方向に並ぶ画像データに対して分析フィルタリングを行う垂直分析フィルタリング処理を行い、ステップS404において画面水平方向に並ぶ画像データに対して分析フィルタリング処理を行う水平分析フィルタリング処理を行う。

【0393】

ステップS405においてウェーブレット変換部410は、分析フィルタリング処理を最終レベルまで行ったか否かを判定し、分解レベルが最終レベルに達していないと判定した場合、処理をステップS403に戻し、現在の分解レベルに対して、ステップS403およびステップS404の分析フィルタリング処理を繰り返す。

【0394】

ステップS405において、分析フィルタリング処理が最終レベルまで行われたと判定した場合、ウェーブレット変換部410は、処理をステップS406に進める。

【0395】

ステップS406において、係数並び替え部413は、ラインブロックA（ピクチャ（インタレース方式の場合フィールド）の上からA番目のラインブロック）の係数を低域から高域の順番に並び替える。エントロピ符号化部415は、ステップS407において、その係数に対してライン毎にエントロピ符号化する。エントロピ符号化が終了すると、エントロピ符号化部415は、ステップS408においてラインブロックAの符号化データを外部に送出する。

## 【0396】

ウェーブレット変換部410は、ステップS409において番号Aの値を「1」インクリメントして次のラインブロックを処理対象とし、ステップS410において、処理対象のピクチャについて、未処理の画像入力ラインが存在するか否かを判定し、存在すると判定した場合、処理をステップS402に戻し、新たな処理対象のラインブロックに対してそれ以降の処理を繰り返す。

## 【0397】

以上のようにステップS402乃至ステップS410の処理が繰り返し実行され、各ラインブロックが符号化される。そして、ステップS410において、未処理の画像入力ラインが存在しないと判定した場合、ウェーブレット変換部410は、そのピクチャに対する符号化処理を終了する。次のピクチャに対しては新たに符号化処理が開始される。

10

## 【0398】

従来のウェーブレット変換の方法の場合、まず、水平分析フィルタリング処理をピクチャ全体に対して行い、次に垂直分析フィルタリング処理をそのピクチャ全体に対して行う。そして得られた低域成分全体に対して同様の水平分析フィルタリング処理と垂直分析フィルタリング処理を順に行う。以上のように、分解レベルが最終レベルに達するまで、分析フィルタリング処理が再帰的に繰り返される。従って、各分析フィルタリング処理の結果をバッファに保持させる必要があるが、その際、バッファは、ピクチャ全体、若しくは、その時点の分解レベルの低域成分全体のフィルタリング結果を保持する必要がある、多大なメモリ容量を必要とすることになる（保持するデータ量が多い）。

20

## 【0399】

また、この場合、ピクチャ内において全てのウェーブレット変換が終了しないと、後段の係数並び替えやエントロピ符号化を行うことができず、遅延時間が増大する。

## 【0400】

これに対して、画像符号化装置401のウェーブレット変換部410の場合、上述したようにラインブロック単位で垂直分析フィルタリング処理および水平分析フィルタリング処理を最終レベルまで連続して行うので、従来の方法と比較して、一度に（同時期に）保持する（バッファリングする）必要のあるデータの量が少なく、用意すべきバッファのメモリ量を大幅に低減させることができる。また、最終レベルまで分析フィルタリング処理が行われることにより、後段の係数並び替えやエントロピ符号化等の処理も行うことができる（つまり、係数並び替えやエントロピ符号化をラインブロック単位で行うことができる）。従って、従来の方法と比較して遅延時間を大幅に低減させることができる。

30

## 【0401】

図27は、図19の画像符号化装置401に対応する画像復号装置の一例の構成を示す。図19の画像符号化装置401のエントロピ符号化部415から出力された符号化データ（図19の符号化データ出力）は、図27の画像復号装置420のエントロピ復号部421に供給され（図27の符号化データ入力）、エントロピ符号を復号され、係数データとされる。係数データは、係数バッファ部422に格納される。ウェーブレット逆変換部423は、係数バッファ部422に格納された係数データを用いて、例えば図23および図24を用いて説明したようにして合成フィルタによる合成フィルタ処理を行い、合成フィルタ処理の結果を再び係数バッファ部422に格納する。ウェーブレット逆変換部423は、この処理を分解レベルに応じて繰り返して、復号された画像データ（出力画像データ）を得る。

40

## 【0402】

図27の画像復号装置420のエントロピ復号部421は、図8の画像復号装置111のエントロピ復号部121および逆量子化部122に対応し、画像復号装置420の係数バッファ部422およびウェーブレット逆変換部423は、画像復号装置111のウェーブレット逆変換部123に対応する。

## 【0403】

つまり、ここではウェーブレット逆変換処理についての説明のため、ウェーブレット逆変換処理の方法と関連性の薄い逆量子化処理（逆量子化部122）の説明を省略している。

50

もちろん、画像符号化装置401において量子化を行うのであれば、エントロピ復号部421がエントロピ復号後に、逆量子化部122と同様の逆量子化（画像符号化装置401における量子化に対応する逆量子化）を行い、得られたウェーブレット係数を、係数バッファ部422（ウェーブレット逆変換部123）に供給するようにしてもよい。このようにすれば、さらに圧縮効果の向上を期待することができる。

【0404】

次に、以上のような画像復号装置420による復号処理全体の具体的な流れの例を図28のフローチャートを参照して説明する。

【0405】

復号処理が開始されると、エントロピ復号部421は、ステップS431において、符号化データを取得し、ステップS432において、ライン毎に符号化データをエントロピ復号する。ステップS433において、係数バッファ部422は、その復号されて得られた係数を保持する。ステップS434においてウェーブレット逆変換部423は、係数バッファ部422に1ラインブロック分の係数が蓄積されたか否かを判定し、蓄積されていないと判定した場合、処理をステップS431に戻し、それ以降の処理を実行させ、係数バッファ部422に1ラインブロック分の係数が蓄積されるまで待機する。

10

【0406】

ステップS434において係数バッファ部422に1ラインブロック分の係数が蓄積されたと判定した場合、ウェーブレット逆変換部423は、処理をステップS435に進め、係数バッファ部422に保持されている係数を1ラインブロック分読み出す。

20

【0407】

そしてその読み出した係数に対して、ウェーブレット逆変換部423は、ステップS436において、画面垂直方向に並ぶ係数に対して合成フィルタリング処理を行う垂直合成フィルタリング処理を行い、ステップS437において、画面水平方向に並ぶ係数に対して合成フィルタリング処理を行う水平合成フィルタリング処理を行い、ステップS438において、合成フィルタリング処理がレベル1（分解レベルの値が「1」のレベル）まで終了したか否か、すなわち、ウェーブレット変換前の状態まで逆変換したか否かを判定し、レベル1まで達していないと判定した場合、処理をステップS436に戻し、ステップS436およびステップS437のフィルタリング処理を繰り返す。

【0408】

30

ステップS438において、レベル1まで逆変換処理が終了したと判定した場合、ウェーブレット逆変換部423は、処理をステップS439に進め、逆変換処理により得られた画像データを外部に出力する。

【0409】

ステップS440において、エントロピ復号部421は、復号処理を終了するか否かを判定し、符号化データの入力が継続しており、復号処理を終了しないと判定した場合、処理をステップS431に戻し、それ以降の処理を繰り返す。また、ステップS440において、符号化データの入力が終了するなどして復号処理を終了すると判定した場合、エントロピ復号部421は、復号処理を終了する。

【0410】

40

従来のウェーブレット逆変換の方法の場合、処理対象の分解レベルの全係数に対して、まず、画面水平方向に水平合成フィルタリング処理を行い、次に画面垂直方向に垂直合成フィルタリング処理を行っていた。つまり、各合成フィルタリング処理の度に、その合成フィルタリング処理の結果をバッファに保持させる必要があるが、その際、バッファは、その時点の分解レベルの合成フィルタリング結果と、次の分解レベルの全係数を保持する必要がある、多大なメモリ容量を必要とすることになる（保持するデータ量が多い）。

【0411】

また、この場合、ピクチャ（インタレース方式の場合フィールド）内において全てのウェーブレット逆変換が終了するまで画像データ出力が行われないので、入力から出力までの遅延時間が増大する。

50

## 【0412】

これに対して、画像復号装置420のウェーブレット逆変換部423の場合、上述したようにラインブロック単位で垂直合成フィルタリング処理および水平合成フィルタリング処理をレベル1まで連続して行うので、従来の方法と比較して、一度に（同時期に）バッファリングする必要のあるデータの量が少なく、用意すべきバッファのメモリ量を大幅に低減させることができる。また、レベル1まで合成フィルタリング処理（ウェーブレット逆変換処理）が行われることにより、ピクチャ内の全画像データが得られる前に（ラインブロック単位で）画像データを順次出力させることができ、従来の方法と比較して遅延時間を大幅に低減させることができる。

## 【0413】

10

なお、図19に示される画像符号化装置401や、図27に示される画像復号装置420の各要素の動作（図26の符号化処理や図28の復号処理）は、例えば図示されないCPU(Central Processing Unit)により、所定のプログラムに従い制御される。プログラムは、例えば図示されないROM(Read Only Memory)に予め記憶される。これに限らず、画像符号化装置や画像復号装置を構成する各要素間でタイミング信号や制御信号を互いにやりとりして、全体として動作させることも可能である。また、画像符号化装置や画像復号装置は、コンピュータ装置上で動作するソフトウェアで実現することも可能である。

## 【0414】

次に、このような画像符号化処理および画像復号処理の他の例について説明する。例えば、上述のシステム（画像符号化装置401および画像復号装置420）において、画像符号化装置401および画像復号装置420の各要素を並列的に動作させ、画像の圧縮符号化および復号処理をより低遅延で行うようにすることができる。

20

## 【0415】

なお、以上において説明した内容と重複する部分の説明は、繁雑さを避けるために省略する。

## 【0416】

図29は、この発明の実施の第2の形態による、画像符号化装置401および画像復号装置420の各要素の一例の並列動作を概略的に示す。この図29は、上述した図25と対応するものである。画像データの入力In-1（図29のA）に対して、エントロピ符号化部415で1回目のウェーブレット変換WT-1が施される（図29のB）。図24を参照し説明したように、この1回目のウェーブレット変換WT-1は、最初の3ラインが入力された時点で開始され、係数C1が生成される。すなわち、画像データIn-1の入力からウェーブレット変換WT-1が開始されるまで、3ライン分の遅延が生じる。

30

## 【0417】

生成された係数データは、係数並び替え用バッファ部412に格納される。以降、入力された画像データに対してウェーブレット変換が施され、1回目の処理が終了すると、そのまま2回目のウェーブレット変換WT-2に処理が移行する。

## 【0418】

2回目のウェーブレット変換WT-2のための画像データIn-2の入力と、当該2回目のウェーブレット変換WT-2の処理と並列的に、係数並び替え部413により3個の、係数C1、係数C4、および係数C5の並び替えOrd-1が実行される（図29のC）。

40

## 【0419】

なお、ウェーブレット変換WT-1の終了から並び替えOrd-1が開始されるまでの遅延は、例えば、並び替え処理を係数並び替え部413に指示する制御信号の伝達に伴う遅延や、制御信号に対する係数並び替え部413の処理開始に要する遅延、プログラム処理に要する遅延といった、装置やシステム構成に基づく遅延であって、符号化処理における本質的な遅延ではない。

## 【0420】

係数データは、並び替えが終了した順に係数並び替え用バッファ部412から読み出され、エントロピ符号化部415に供給され、エントロピ符号化EC-1が行われる（図29のD）。こ

50

のエントロピ符号化EC-1は、3個の、係数C1、係数C4、および係数C5の、全ての並び替えの終了を待たずに開始することができる。例えば、最初に出力される係数C5による1ラインの並び替えが終了した時点で、当該係数C5に対するエントロピ符号化を開始することができる。この場合、並び替えOrd-1の処理開始からエントロピ符号化EC-1の処理開始までの遅延は、1ライン分となる。

#### 【0421】

エントロピ符号化部415によるエントロピ符号化EC-1が終了した符号化データは、何らかの伝送路を介して画像復号装置420に伝送される（図29のE）。符号化データが伝送される伝送路としては、例えばインターネットなどの通信ネットワークが考えられる。この場合、符号化データは、IP(Internet Protocol)により伝送される。これに限らず、符号化データの伝送路としては、USB(Universal Serial Bus)やIEEE1394(Institute Electrical and Electronics Engineers 1394)といった通信インタフェースや、IEEE802.11規格などに代表される無線通信も考えられる。

10

#### 【0422】

画像符号化装置401に対して、1回目の処理による7ライン分の画像データ入力に続けて、画面上の下端のラインまで画像データが順次、入力される。画像符号化装置401では、画像データの入力In-n（nは2以上）に伴い、上述したようにして、4ライン毎にウェーブレット変換WT-n、並び替えOrd-nおよびエントロピ符号化EC-nを行う。画像符号化装置401における最後の回の処理に対する並び替えOrdおよびエントロピ符号化ECは、6ラインに対して行われる。これらの処理は、画像符号化装置401において、図29のA乃至図29のDに例示されるように、並列的に行われる。

20

#### 【0423】

画像符号化装置401によるエントロピ符号化EC-1により符号化された符号化データが、画像復号装置420に伝送路を介して伝送され、エントロピ復号部421に供給される。エントロピ復号部421は、供給された、エントロピ符号化EC-1により符号化された符号化データに対して、順次、エントロピ符号の復号iEC-1を行い、係数データを復元する（図29のF）。復元された係数データは、順次、係数バッファ部422に格納される。ウェーブレット逆変換部423は、係数バッファ部422にウェーブレット逆変換が行えるだけ係数データが格納されたら、係数バッファ部422から係数データを読み出して、読み出された係数データを用いてウェーブレット逆変換iWT-1を行う（図29のG）。

30

#### 【0424】

図24を参照して説明したように、ウェーブレット逆変換部423によるウェーブレット逆変換iWT-1は、係数C4および係数C5が係数バッファ部422に格納された時点で開始することができる。したがって、エントロピ復号部421による復号iEC-1が開始されてからウェーブレット逆変換部423によるウェーブレット逆変換iWT-1が開始されるまでの遅延は、2ライン分となる。

#### 【0425】

ウェーブレット逆変換部423において、1回目のウェーブレット変換による3ライン分のウェーブレット逆変換iWT-1が終了すると、ウェーブレット逆変換iWT-1で生成された画像データの出力Out-1が行われる（図29のH）。出力Out-1では、図24および図25を用いて説明したように、第1ライン目の画像データが出力される。

40

#### 【0426】

画像復号装置420に対して、画像符号化装置401における1回目の処理による3ライン分の符号化された係数データの入力に続けて、エントロピ符号化EC-n（nは2以上）により符号化された係数データが順次、入力される。画像復号装置420では、入力された係数データに対して、上述したようにして、4ライン毎にエントロピ復号iEC-nおよびウェーブレット逆変換iWT-nを行い、ウェーブレット逆変換iWT-nにより復元された画像データの出力Out-nを順次、行う。画像符号化装置の最後の回に対応するエントロピ復号iECおよびウェーブレット逆変換iWTは、6ラインに対して行われ、出力Outは、8ラインが出力される。これらの処理は、画像復号装置において、図29のF乃至図29のHに例示されるように、並列的に

50

われる。

#### 【0427】

上述のようにして、画面上部から下部の方向に順番に、画像符号化装置401および画像復号装置420における各処理を並列的に行うことで、画像圧縮処理および画像復号処理をより低遅延で行うことが可能となる。

#### 【0428】

図29を参照して、 $5 \times 3$ フィルタを用いて分解レベル=2までウェーブレット変換を行った場合の、画像入力から画像出力までの遅延時間を計算してみる。第1ライン目の画像データが画像符号化装置401に入力されてから、この第1ライン目の画像データが画像復号装置420から出力されるまでの遅延時間は、下記の各要素の総和となる。なお、ここでは、伝送路における遅延や、装置各部の実際の処理タイミングに伴う遅延などの、システムの構成により異なる遅延は、除外している。

#### 【0429】

(1) 最初のライン入力から7ライン分のウェーブレット変換WT-1が終了するまでの遅延 $D_{WT}$

(2) 3ライン分の計数並び替えOrd-1に伴う時間 $D_{Ord}$

(3) 3ライン分のエントロピ符号化EC-1に伴う時間 $D_{EC}$

(4) 3ライン分のエントロピ復号iEC-1に伴う時間 $D_{iEC}$

(5) 3ライン分のウェーブレット逆変換iWT-1に伴う時間 $D_{iWT}$

#### 【0430】

図29を参照して、上述の各要素による遅延の計算を試みる。(1)の遅延 $D_{WT}$ は、10ライン分の時間である。(2)の時間 $D_{Ord}$ 、(3)の時間 $D_{EC}$ 、(4)の時間 $D_{iEC}$ 、および(5)の時間 $D_{iWT}$ は、それぞれ3ライン分の時間である。一方、画像符号化装置401において、並び替えOrd-1が開始されてから1ライン後には、エントロピ符号化EC-1を開始することができる。同様に、画像復号装置420において、エントロピ復号iEC-1が開始されてから2ライン後には、ウェーブレット逆変換iWT-1を開始することができる。また、エントロピ復号iEC-1は、エントロピ符号化EC-1で1ライン分の符号化が終了した時点で処理を開始することができる。

#### 【0431】

したがって、この図29の例では、画像符号化装置に第1ライン目の画像データが入力されてから、画像復号装置から当該第1ライン目の画像データが出力されるまでの遅延時間は、 $10+1+1+2+3=17$ ライン分となる。

#### 【0432】

遅延時間について、より具体的な例を挙げて考察する。入力される画像データがHDTV(High Definition Television)のインタレースビデオ信号の場合、例えば1920画素 $\times$ 1080ラインの解像度で1フレームが構成され、1フィールドは、1920画素 $\times$ 540ラインとなる。したがって、フレーム周波数を30Hzとした場合、1フィールドの540ラインが16.67msec (= 1sec/60フィールド)の時間に、画像符号化装置401に入力されることになる。

#### 【0433】

したがって、7ライン分の画像データの入力に伴う遅延時間は、0.216msec (= 16.67msec $\times$ 7/540ライン)であり、例えば1フィールドの更新時間に対して非常に短い時間となる。また、上述した(1)の遅延 $D_{WT}$ 、(2)の時間 $D_{Ord}$ 、(3)の時間 $D_{EC}$ 、(4)の時間 $D_{iEC}$ 、および(5)の時間 $D_{iWT}$ の総和についても、処理対象のライン数が少ないため、遅延時間が非常に短縮される。各処理を行う要素をハードウェア化すれば、処理時間をさらに短縮することも可能である。

#### 【0434】

さらに、他の例について説明する。上述した説明では、画像符号化装置401において、ウェーブレット変換を行った後に係数データの並び替えを行っていた。これに対して、以下においては、係数データの並び替えを、エントロピ符号化の後に行うようにする場合について説明する。この場合の画像符号化装置では、入力された画像データに対してウェー

10

20

30

40

50

ウェーブレット変換を行い生成された係数に対してエントロピ符号化を施し、エントロピ符号化されたデータに対して並び替え処理を行う。このように、係数データの並び替えをエントロピ符号化の後で行うことで、係数並び替えバッファにおいて必要とされる記憶容量を抑えることができる。

#### 【0435】

例えば、入力画像データのビット精度が8ビットの場合、ウェーブレット変換を複数レベル分解まで行くと、生成される係数データのビット精度が例えば12ビット程度となる。エントロピ符号化処理の前に係数並び替え処理を行う場合、係数並び替え用バッファ部は、このビット精度12ビットの係数データを、所定ライン数分、格納する必要がある。ウェーブレット変換で生成された係数データをエントロピ符号化した後に並び替え処理するよう

10

#### 【0436】

図30は、この場合の画像符号化装置の一例の構成を示す。なお、図30において、上述した図19と共通する部分には同一の符号を付し、詳細な説明を省略する。

#### 【0437】

入力画像データは、画像符号化装置430の途中計算用バッファ部411に一旦格納される。ウェーブレット変換部410は、途中計算用バッファ部411に格納された画像データに対して、実施の第1の形態で既に説明したようにして、ウェーブレット変換を所定に施す。ウェーブレット変換で生成された係数データは、エントロピ符号化部415に供給される。エントロピ符号化部415は、レート制御部414と連動的に動作し、出力される圧縮符号化データのビットレートが略一定値となるように制御され、供給された係数データに対してエントロピ符号化処理を行う。つまり、エントロピ符号化部415は、係数の順序に関わらず、取得した係数を、その取得した順序で同様に符号化する。

20

#### 【0438】

ウェーブレット変換により生成された係数データがエントロピ符号化部415でエントロピ符号化された符号化データは、符号並び替え用バッファ部431に一旦格納される。符号並び替え部432は、符号並び替え用バッファ部431に並び替える符号化データが格納され次第、符号並び替え用バッファ部431から符号化データを並び替えて読み出す。実施の第1の形態で既に説明したように、ウェーブレット変換部410で生成された係数データは、高域成分から低域成分の順に、画面の上端側から下端側に向けて生成される。復号側において低遅延で画像データの出力を行うためには、符号並び替え用バッファ部431に格納された符号化データを、ウェーブレット変換による係数データの低域成分から高域成分の順に並び替えて読み出す。

30

#### 【0439】

符号並び替え用バッファ部431から読み出された符号化データが出力の符号化データとして、例えば伝送路に送出される。

#### 【0440】

なお、この画像符号化装置430で符号化され出力されたデータは、図27を用いて既に説明した画像復号装置420により、画像符号化装置401において符号化されたデータを復号する場合と同様に復号することができる。すなわち、例えば伝送路を介して画像復号装置420に入力された符号化データは、エントロピ復号部421でエントロピ符号の復号がなされ、係数データが復元される。復元された係数データは、係数バッファ部422に順次、格納される。ウェーブレット逆変換部423は、係数バッファ部422に格納された係数データに対してウェーブレット逆変換を施し、画像データを出力する。

40

#### 【0441】

さらに、他の例について説明する。以上においては、ウェーブレット変換で生成された係数データの並び替え処理を、図31に一例が示されるように、画像符号化装置側で行うように説明した。ここでは、ウェーブレット変換により生成された係数データの並び替え処理を、図32で一例が示されるように、画像復号装置側で行う場合について説明する。

50



## 【0442】

ウェーブレット変換で生成された係数データを並び替える処理では、上述したように、係数並び替え用バッファの記憶容量として比較的大容量が必要となると共に、係数並び替えの処理自体にも、高い処理能力が要求される。この場合でも、画像符号化装置側の処理能力がある程度以上高い場合には、上述した各説明の場合のように、画像符号化装置側で係数並び替え処理を行っても、何ら問題は生じない。

## 【0443】

ここで、携帯電話端末やPDA(Personal Digital Assistant)といった所謂モバイル端末などの、比較的处理能力の低い機器に画像符号化装置が搭載される場合について考える。例えば、近年では、携帯電話端末に対して撮像機能を付加した製品が広く普及している（カメラ機能付き携帯電話端末と呼ぶ）。このようなカメラ機能付き携帯電話端末で撮像された画像データをウェーブレット変換およびエントロピ符号化により圧縮符号化し、無線あるいは有線通信を介して伝送することが考えられる。

10

## 【0444】

このような例えばモバイル端末は、CPUの処理能力も限られ、また、メモリ容量にもある程度の上限がある。そのため、上述したような係数並び替えに伴う処理の負荷などは、無視できない問題となる。

## 【0445】

そこで、図32に一例が示されるように、並び替え処理を画像復号装置側に組み入れることで、画像符号化装置側の負荷が軽くなり、画像符号化装置をモバイル端末などの比較的处理能力が低い機器に搭載することが可能となる。

20

## 【0446】

図33は、この場合の画像符号化装置の一例の構成を示す。なお、この図33において、上述の図19と共通する部分には同一の符号を付して、詳細な説明を省略する。

## 【0447】

この図33に示される画像符号化装置441の構成は、上述の図19で示した画像符号化装置401の構成に対して係数並び替え部413および係数並び替え用バッファ部412を除去した構成となっている。すなわち、この実施の第4の形態では、画像符号化装置441としては、従来から用いられる、ウェーブレット変換部410、途中計算用バッファ部411、エントロピ符号化部415およびレート制御部414を組み合わせた構成を適用することが可能である。

30

## 【0448】

入力された画像データは、途中計算用バッファ部411に一時的に溜め込まれる。ウェーブレット変換部410は、途中計算用バッファ部411に溜め込まれた画像データに対してウェーブレット変換を施し、生成された係数データを、係数データの生成順に順次、エントロピ符号化部415に供給する。すなわち、エントロピ符号化部415に対して、ウェーブレット変換の順序に従い高域成分から低域成分の順に、生成された係数データが供給される。エントロピ符号化部415は、供給された係数に対して、レート制御部414により出力データのビットレートを制御されながらエントロピ符号化を施す。エントロピ符号化部415から、ウェーブレット変換により生成された係数データがエントロピ符号化された符号化データが出力される。

40

## 【0449】

図34は、この場合の画像復号装置の一例の構成を示す。なお、この図34において、上述の図27と共通する部分には同一の符号を付し、詳細な説明を省略する。

## 【0450】

図33で説明した画像符号化装置441のエントロピ符号化部415から出力された符号化データは、図34の画像復号装置442のエントロピ復号部421に供給され、エントロピ符号を復号され係数データとされる。係数データは、係数バッファ部422を介して係数並び替え用バッファ部443に格納される。ウェーブレット逆変換部423は、係数並び替え用バッファ部443に係数データの並び替えが可能となるまで係数データが蓄積されると、係数並び替え用バッファ部443に格納された係数データを、低域成分から高域成分の順に並び替えて読み

50

出し、読み出された順に係数データを用いてウェーブレット逆変換処理を行う。5×3フィルタを用いる場合は、上述の図32で示したようになる。

【0451】

すなわち、ウェーブレット逆変換部423は、例えば1フレームの先頭からの処理であれば、係数並び替え用バッファ部443にエントロピ符号の復号がなされた係数C1、係数C4、および係数C5が格納された時点で、係数並び替え用バッファ部443から係数データを読み出し、ウェーブレット逆変換処理を行う。ウェーブレット逆変換部423でウェーブレット逆変換を施されたデータは、順次、出力画像データとして出力される。

【0452】

なお、この場合でも図29を用いて既に説明した場合と同様に、画像符号化装置441における各要素の処理と、伝送路に対する符号化データの伝送と、画像復号装置442における各要素の処理とが並列的に実行することができる。

10

【0453】

次に、上述した画像符号化装置と画像復号装置の間での符号化データの授受について説明する。図35は、その符号化データの授受の様子を説明する模式図である。図35に示される例の場合においても、上述した他の実施の形態と同様に、画像データは、ラインブロック毎に、所定のライン数分だけ入力されながらウェーブレット変換される（サブバンド451）。そして、所定のウェーブレット変換分解レベルまで達した際に、最低域サブバンドから最高域サブバンドまでの係数ラインが、生成された順序とは逆に、つまり低域から高域の順番に並び替えられる。

20

【0454】

図35のサブバンド451において、斜め線、縦線、および波線の模様分けされた部分は、それぞれ異なるラインブロックである（矢印で示されるように、サブバンド451の白ヌキ部分も同様にラインブロック毎に分割して処理される）。並び替えられた後のラインブロックの係数が上述した様にエントロピ符号化され、符号化データが生成される。

【0455】

ここで、例えば画像符号化装置が符号化データをそのまま送出すると、画像復号装置が各ラインブロックの境界を識別することが困難な（若しくは煩雑な処理が必要になる）場合がある。そこで、本実施の形態においては、画像符号化装置が、符号化データに例えばラインブロック単位でヘッダを付加し、ヘッダと符号化データとから成るパケットとして送出するようにする。

30

【0456】

つまり、画像符号化装置は、図35に示されるように、1番目のラインブロック（Lineblock-1）の符号化データ（エンコードデータ）を生成すると、それをパケット化し、送信パケット461として画像復号装置に送出する。画像復号装置は、そのパケットを受信すると（受信パケット471）、その符号化データを復号（デコード）する。

【0457】

同様に、画像符号化装置は、2番目のラインブロック（Lineblock-2）の符号化データを生成すると、それをパケット化し、送信パケット462として画像復号装置に送出する。画像復号装置は、そのパケットを受信すると（受信パケット472）、その符号化データを復号（デコード）する。さらに同様に、画像符号化装置は、3番目のラインブロック（Lineblock-3）の符号化データを生成すると、それをパケット化し、送信パケット463として画像復号装置に送出する。画像復号装置は、そのパケットを受信すると（受信パケット473）、その符号化データを復号（デコード）する。

40

【0458】

画像符号化装置および画像復号装置は、以上のような処理を、X番目の最終ラインブロック（Lineblock-X）まで繰り返す（送信パケット464、受信パケット474）。以上のようにして画像復号装置において復号画像481が生成される。

【0459】

図36にヘッダの構成例を示す。上述したようにパケットはヘッダ（Header）491と符号

50

化データにより構成されるが、そのヘッダ491には、ラインブロックの番号 (NUM) 493と符号化データ長 (LEN) 494の記述が含まれている。

【0460】

画像復号装置は、受信した符号化データに付加されたヘッダに含まれるこれらの情報を読み取ることにより、各ラインブロックの境界を容易に識別することができ、復号処理の負荷や処理時間を低減させることができる。

【0461】

なお、図36に示されるように、さらに、ラインブロックを構成するサブバンド毎の量子化ステップサイズ (1乃至 N) 492の記述を付加するようにしてもよい。これにより、画像復号装置は、サブバンド毎の逆量子化を行うことができ、よりきめ細かな画質制御を行うことが出来る。

10

【0462】

また、画像符号化装置および画像復号装置は、上述したような符号化、パケット化、パケットの送受信、および復号等の各処理を、第4の実施の形態において説明したように、ラインブロック毎に、同時並行的に (パイプライン化して) 実行するようにしてもよい。

【0463】

このようにすることにより、画像復号装置において画像出力が得られるまでの遅延時間を大幅に低減させることができる。図35においては一例として、インタレース動画 (60フィールド/秒) での動作例を示している。この例において、1フィールドの時間は、 $1\text{秒} \div 60 = \text{約}16.7\text{msec}$ であるが、同時並行的に各処理を行うようにすることにより、約5msecの遅延時間で画像出力が得られるようにすることが出来る。

20

【0464】

以上において各例を説明したように、ラインブロック毎にウェーブレット変換 (またはウェーブレット逆変換) を行うようにすることにより、画像符号化装置 (または画像復号装置) は、さらに高速に符号化処理 (または復号処理) を行うことができる。

【0465】

換言するに、エントロピ符号化部は、ラインブロック毎に変換するウェーブレット変換部からの係数の出力順の特徴 (つまり連続する係数の有効桁数が似ていること) を生かして発生符号量を低減させることができる。また、係数の並び替えを行う場合であっても、ウェーブレット変換部がラインブロック単位でウェーブレット変換を行うので、連続する係数の有効桁数が似ているという特徴には大きな影響を与えることがなく、エントロピ符号化処理の発生符号量は大きく変化しない。

30

【0466】

以上のように、上述したウェーブレット変換処理は、本発明のエントロピ符号化処理と、処理される係数データの特徴や期待される効果が似ており、互いに親和性が高い。従って、一般的な従来のウェーブレット変換方法を適用する場合よりも、上述したウェーブレット変換処理を適用する場合の方が、画像符号化処理全体においてより大きな効果を得ることを期待することができる。

【0467】

そして、上述したように、このウェーブレット変換処理は、多様な構成により実現可能であり、本発明のエントロピ符号化処理やエントロピ復号処理 (画像符号化装置や画像復号装置) を、多種のシステムに適用することができる。

40

【0468】

以下に具体的な適用例を説明する。最初に、上述した画像符号化装置および画像復号装置を、ディジタルトライアックスシステムに適用する場合について説明する。

【0469】

トライアックスシステムは、テレビジョン放送局や制作スタジオなどにおいて、スタジオ収録や中継などの際に、ビデオカメラと、カメラコントロールユニットやスイッチャとを接続する1本の同軸ケーブルで、映像信号、音声信号、送り返し (リターン) の映像信号、同期信号など複数の信号を重畳させて送信すると共に、電源の供給も行うようにした

50

システムである。

【0470】

従来のトライアックスシステムは、アナログ信号を用いて上述の各信号の伝送を行うようにした例が大部分であった。一方、近年では、システム全体のデジタル化に伴い、放送局内などで用いるトライアックスシステムのデジタル化が進んでいる。

【0471】

既存のデジタルトライアックスシステムでは、トライアックスケーブルを介して伝送されるデジタルビデオ信号は、非圧縮のビデオ信号であった。これは、特に放送局などにおいては信号の遅延時間に対する要求スペックが厳しく、基本的に、撮像から例えばモニタ出力までの遅延時間が1フィールド(16.67msec)以内であることが要求されるためである。高圧縮率および高画質を実現したMPEG2(Moving Pictures Experts Group 2)やMPEG4といった圧縮符号化方式は、ビデオ信号の圧縮符号化や圧縮ビデオ信号の復号に際して数フレーム分の時間を要し、遅延時間が大きいいため、トライアックスシステムに採用されることはなかった。

10

【0472】

この発明による画像符号化および画像復号方法は、上述したように、画像データの入力から出力画像が得られるまでの遅延時間が1フィールド時間以内、例えば数ライン乃至数十ラインと極めて短く、デジタルトライアックスシステムに対して用いて好適なものである。

【0473】

20

図37は、本発明を適用したデジタルトライアックスシステムの一例の構成を示す。送信ユニット500とカメラ制御部502とがトライアックスケーブル(同軸ケーブル)501を介して接続される。送信ユニット500からカメラ制御部502に対する、実際に放映されたり素材として用いられたりするデジタルビデオ信号およびデジタルオーディオ信号(以下、本線信号と呼ぶ)の送出、カメラ制御部502からビデオカメラ部503に対する、インカム用のオーディオ信号およびリターン用のデジタルビデオ信号の送出が、トライアックスケーブル501を介して行われる。

【0474】

送信ユニット500は、例えば図示されないビデオカメラ装置に内蔵される。これに限らず、送信ユニット500がビデオカメラ装置に対する外部装置として、ビデオカメラ装置と所定に接続されて用いられるものとしてもよい。また、カメラ制御部502は、例えば一般的にCCU(Camera Control Unit)と呼ばれる装置である。

30

【0475】

なお、デジタルオーディオ信号については、この発明の主旨と関わりが少ないので、複雑さを避けるための説明を省略する。

【0476】

ビデオカメラ部503は、例えば図示されないビデオカメラ装置内に構成され、レンズ、フォーカス機構、ズーム機構、アイリス調整機構などを有する光学系550を介して入射された被写体からの光を、CCD(Charge Coupled Device)などからなる図示されない撮像素子で受光する。撮像素子は、受光された光を光電変換で電気信号に変換し、さらに所定の信号処理を施してベースバンドのデジタルビデオ信号を出力する。このデジタルビデオ信号は、例えばHD-SDI(High Definition-Serial Data Interface)のフォーマットにマッピングして出力される。

40

【0477】

また、ビデオカメラ部503は、モニタ用に用いられる表示部551と、外部と音声によるやりとりを行うためのインカム552が接続される。

【0478】

送信ユニット500は、ビデオ信号符号化部510およびビデオ信号復号部511、デジタル変調部512およびデジタル復調部513、アンプ514およびアンプ515、並びに、ビデオ分離/合成部516を有する。

50

## 【 0 4 7 9 】

送信ユニット500において、ビデオカメラ部503から、例えばHD-SDIのフォーマットにマッピングされたベースバンドのデジタルビデオ信号が供給される。このデジタルビデオ信号は、ビデオ信号符号化部510で圧縮符号化され、符号化ストリームとされてデジタル変調部512に供給される。デジタル変調部512は、供給された符号化ストリームを、トライアックスケーブル501を介した伝送に適した形式の信号に変調して出力する。デジタル変調部512から出力された信号は、アンプ514を介してビデオ分離 / 合成部516に供給される。ビデオ分離 / 合成部516は、供給された信号をトライアックスケーブル501に送出する。この信号は、トライアックスケーブル501を介してカメラ制御部502に受信される。

10

## 【 0 4 8 0 】

カメラ制御部502から出力された信号が、トライアックスケーブル501を介して送信ユニット500に受信される。受信された信号は、ビデオ分離 / 合成部516に供給され、デジタルビデオ信号の部分とその他の信号の部分とが分離される。受信信号のうちデジタルビデオ信号の部分は、アンプ515を介してデジタル復調部513に供給され、カメラ制御部502側でトライアックスケーブル501を介した伝送に適した形式の信号に変調された信号を復調し、符号化ストリームを復元する。

## 【 0 4 8 1 】

符号化ストリームは、ビデオ信号復号部511に供給され、圧縮符号を復号され、ベースバンドのデジタルビデオ信号とされる。この復号されたデジタルビデオ信号は、HD-SDIのフォーマットにマッピングされて出力され、リターン用のデジタルビデオ信号としてビデオカメラ部503に供給される。このリターン用のデジタルビデオ信号は、ビデオカメラ部503に接続される表示部551に供給され、撮影者のためのモニタなどに利用される。

20

## 【 0 4 8 2 】

カメラ制御部502は、ビデオ分離 / 合成部520、アンプ521およびアンプ522、フロントエンド部523、デジタル復調部524およびデジタル変調部525、並びに、ビデオ信号復号部526およびビデオ信号符号化部527を有する。

## 【 0 4 8 3 】

送信ユニット500から出力された信号が、トライアックスケーブル501を介してカメラ制御部502に受信される。受信された信号は、ビデオ分離 / 合成部520に供給される。ビデオ分離 / 合成部520は、供給された信号を、アンプ521およびフロントエンド部523を介してデジタル復調部524に供給する。なお、フロントエンド部523は、入力信号のゲインを調整するゲイン制御部や、入力信号に対して所定のフィルタ処理を施すフィルタ部などを有する。

30

## 【 0 4 8 4 】

デジタル復調部524は、送信ユニット500側でトライアックスケーブル501を介した伝送に適した形式の信号に変調された信号を復調し、符号化ストリームを復元する。この符号化ストリームは、ビデオ信号復号部526に供給され、圧縮符号を復号され、ベースバンドのデジタルビデオ信号にとされる。この復号されたデジタルビデオ信号は、HD-SDIのフォーマットにマッピングされて出力され、本線信号として外部に出力される。

40

## 【 0 4 8 5 】

外部からカメラ制御部502に対して、リターン用のデジタルビデオ信号と、デジタルオーディオ信号とが供給される。デジタルオーディオ信号は、例えば、撮影者のインカム552に供給され、外部から撮影者に対する音声による指示を伝達するのに用いられる。

## 【 0 4 8 6 】

リターン用のデジタルビデオ信号は、ビデオ信号符号化部527に供給されて圧縮符号化され、デジタル変調部525に供給される。デジタル変調部525は、供給された符号化ストリームを、トライアックスケーブル501を介した伝送に適した形式の信号に変調して

50

出力する。ディジタル変調部525から出力された信号は、フロントエンド部523およびアンプ522を介してビデオ分離 / 合成部520に供給される、ビデオ分離 / 合成部520は、この信号を他の信号と多重化し、トライアックスケーブル501に送出する。この信号は、トライアックスケーブル501を介してビデオカメラ部503に受信される。

【0487】

この発明の実施の第7の形態では、上述したビデオ信号符号化部510およびビデオ信号符号化部527、並びに、ビデオ信号復号部511およびビデオ信号復号部526に対して、上述した各実施の形態で説明した画像符号化装置および画像復号装置をそれぞれ適用する。

【0488】

特に、画像符号化装置および画像復号装置における各要素の処理を並列的に行うようにした、発明の実施の第2の形態は、ビデオカメラ部503で撮影された映像がカメラ制御部502から出力される際の遅延や、外部から供給されカメラ制御部502からビデオカメラ部503に送信されるリターン用のディジタルビデオ信号の遅延を低く抑えることができ、この発明の実施の第7の形態に用いて好適である。

【0489】

また、図37に例示したシステムの場合、送信ユニット500およびカメラ制御部502のそれぞれにおいて、信号処理能力やメモリ容量を、適宜、設定することができると考えられるので、係数データの並び替え処理を行う位置は、送信ユニット500側およびカメラ制御部502側の何れであってもよいし、エン트로ピ符号化を行う位置も同様に、並び替え処理の前後の何れであってもよい。

【0490】

すなわち、送信ユニット500側において、ビデオ信号符号化部510は、供給されたディジタルビデオ信号に対して、この発明による方法に従いウェーブレット変換およびエン트로ピ符号化を施し、符号化ストリームを出力する。上述したように、ビデオ信号符号化部510は、ウェーブレット変換に用いるフィルタのタップ数およびウェーブレット変換の分解レベル数に応じたライン数が入力されると、ウェーブレット変換を開始する。そして、図23、図24および図29などを用いて説明したように、画像符号化装置および画像復号装置において、各要素に必要な係数データが蓄積されると、順次、各要素による処理が行われる。1フレームまたは1フィールドの下端のラインまで処理が終了したら、次の1フレームまたは1フィールドの処理が開始される。

【0491】

カメラ制御部502側から送信ユニット500側に、リターン用のディジタルビデオ信号を送信する場合も、同様である。すなわち、カメラ制御部502側において、ビデオ信号符号化部527は、外部から供給されたリターン用のディジタルビデオ信号に対して、この発明による方法に従いウェーブレット変換およびエン트로ピ符号化を施し、符号化ストリームを出力する。

【0492】

ここで、リターン用のディジタルビデオ信号は、本線信号のディジタルビデオ信号よりも画質が低くても構わないとされている場合が多い。そこで、ビデオ信号符号化部527において、符号化時のビットレートを下げるとよい。例えば、ビデオ信号符号化部527において、レート制御部414により、エン트로ピ符号化部415でのエン트로ピ符号化処理をより低ビットレートになるまで行うように制御する。また例えば、カメラ制御部502側では、ビデオ信号符号化部527においてウェーブレット変換部410でより高い分解レベルまで変換処理を行い、送信ユニット500側では、ビデオ信号復号部511のウェーブレット逆変換部423におけるウェーブレット逆変換を、より低い分解レベルまでに止める方法も考えられる。カメラ制御部502側のビデオ信号符号化部527における処理は、この例に限らず、ウェーブレット変換における分解レベルをより低く抑えて変換処理による負担を軽減することも考えられる。

【0493】

次に、本発明の画像符号化装置および画像復号装置を、符号化された符号化データの伝

10

20

30

40

50

送を、無線通信を用いて行うシステムに適用した場合の例について説明する。図38は、本発明を適用した無線伝送システムの一例の構成を示す。なお、この図38の例では、ビデオ信号はビデオカメラまたは送信ユニット600（以下、送信ユニット600と略称する）側から受信装置601側に、一方向的に送信される。オーディオ信号およびその他の信号は、送信ユニット600と受信装置601との間で双方向の通信が可能である。

【0494】

送信ユニット600は、例えばビデオカメラ部602を有する図示されないビデオカメラ装置に内蔵される。これに限らず、送信ユニット600がビデオカメラ部602を有するビデオカメラ装置に対する外部装置として、ビデオカメラ装置と所定に接続されて用いられるものとしてもよい。

10

【0495】

ビデオカメラ部602は、例えば所定の光学系と、例えばCCDからなる撮像素子と、撮像素子から出力された信号をデジタルビデオ信号として出力する信号処理部とを有する。ビデオカメラ部602から、例えばHD-SDIのフォーマットにマッピングされてデジタルビデオ信号が出力される。これはこの例に限らず、ビデオカメラ部602から出力されるデジタルビデオ信号は、他のフォーマットでもよい。

【0496】

送信ユニット600は、ビデオ信号符号化部610、デジタル変調部611および無線モジュール部612を有する。送信ユニット600において、ビデオカメラ部602から、ベースバンドのデジタルビデオ信号が例えばHD-SDIのフォーマットにマッピングされて出力される。このデジタルビデオ信号は、ビデオ信号符号化部610でこの発明に係る圧縮符号化方法により、ウェーブレット変換およびエントロピ符号化により圧縮符号化され、符号化ストリームとされてデジタル変調部611に供給される。デジタル変調部611は、供給された符号化ストリームを、無線通信を行うために適した形式の信号にデジタル変調して出力する。

20

【0497】

また、デジタル変調部611には、デジタルオーディオ信号やその他の信号、例えば所定のコマンドやデータも供給される。例えば、ビデオカメラ部602は、マイクロホンを含み、集音された音声を音声信号に変換し、さらに当該音声信号をA/D変換してデジタルオーディオ信号として出力する。また、ビデオカメラ部602は、所定のコマンドやデータを出力できるようになっている。コマンドやデータは、ビデオカメラ部602の内部で発生するようにしても良いし、ビデオカメラ部602に操作部を設け、当該操作部に対するユーザの操作に応じてコマンドやデータを生成するようにしてもよい。ビデオカメラ部602に対してコマンドやデータを入力する入力装置を接続するようにしてもよい。

30

【0498】

デジタル変調部611は、これらデジタルオーディオ信号やその他の信号をデジタル変調し、出力する。デジタル変調部611から出力されたデジタル変調信号は、無線モジュール部612に供給されアンテナ613から電波として無線送信される。

【0499】

なお、無線モジュール部612は、受信装置601側からの自動再送要求 (ARQ : Auto Repeat Request) を受信すると、デジタル変調部611に対してこのARQを通知し、データの再送を要求する。

40

【0500】

アンテナ613から送信された電波は、受信装置601側のアンテナ620で受信され、無線モジュール部621に供給される。無線モジュール部621は、受信された電波に基づくデジタル変調信号をフロントエンド部622に供給する。フロントエンド部622は、供給されたデジタル変調信号に対して例えばゲイン制御といった所定の信号処理を施して、デジタル復調部623に供給する。デジタル復調部623は、供給されたデジタル変調信号を復調し、符号化ストリームを復元する。

【0501】

50

ディジタル復調部623で復元された符号化ストリームは、ビデオ信号復号部624に供給され、この発明に係る復号方法により圧縮符号を復号され、ベースバンドのディジタルビデオ信号とされる。この復号されたディジタルビデオ信号は、例えばHD-SDIのフォーマットにマッピングされて出力される。

【0502】

ディジタル復調部623には、送信ユニット600側でディジタル変調され送信された、ディジタルオーディオ信号やその他の信号も供給される。ディジタル復調部623は、これらディジタルオーディオ信号やその他の信号がディジタル変調された信号を復調し、ディジタルオーディオ信号やその他の信号を復元して出力する。

【0503】

また、フロントエンド部622は、無線モジュール部621から供給された受信信号に対して所定の方法でエラー検出を行い、例えば誤ったフレームが受信されたといったエラーが検出されると、ARQを出力する。ARQは、無線モジュール部621に供給され、アンテナ620から送信される。

【0504】

このような構成において、送信ユニット600を例えばビデオカメラ部602を有する比較的小型のビデオカメラ装置に内蔵させ、受信装置601にはモニタ装置を接続し、ビデオ信号復号部624から出力されたディジタルビデオ信号をモニタ装置に供給する。受信装置601に対して、送信ユニット600の内蔵されたビデオカメラ装置が無線モジュール部612から送信される電波の到達範囲内にあれば、ビデオカメラ装置で撮影された映像を、低遅延、例えば1フィールドまたは1フレーム時間以内の遅延でモニタ装置により見ることができる。

【0505】

なお、図38では、送信ユニット600と受信装置601との間の通信を、無線通信を用いて行い、ビデオ信号を、無線通信を介して伝送するようにしているが、これはこの例に限定されない。例えば、送信ユニット600と受信装置601は、インターネットなどのネットワークを介して接続するようにしてもよい。この場合、送信ユニット600側の無線モジュール部612および受信装置601側の無線モジュール部621は、それぞれIP(Internet Protocol)を用いた通信が可能な通信インタフェースとされる。

【0506】

このシステムは、さらに様々な応用が考えられる。例えば、このシステムは、テレビジョン会議システムに応用することができる。例えば、USB(Universal Serial Bus)接続が可能な簡易的なビデオカメラ装置をパーソナルコンピュータといったコンピュータ装置に接続すると共に、コンピュータ装置側にビデオ信号符号化部610およびビデオ信号復号部624を搭載する。コンピュータ装置に搭載されるビデオ信号符号化部610およびビデオ信号復号部624は、ハードウェアで構成してもよいし、コンピュータ装置上で動作するソフトウェアとして実現することも可能である。

【0507】

例えば、会議に参加するメンバそれぞれに、コンピュータ装置と当該コンピュータ装置に接続されるビデオカメラ装置が用意され、コンピュータ装置が例えばテレビジョン会議システムのサービスを提供するサーバ装置に、有線および/または無線によるネットワークを介して接続される。ビデオカメラ装置から出力されたビデオ信号は、USBケーブルを介してコンピュータ装置に供給され、コンピュータ装置内のビデオ信号符号化部610で、この発明に係る符号化処理を施される。コンピュータ装置は、ビデオ信号が符号化された符号化ストリームを、ネットワークを介してサーバ装置などに送信する。

【0508】

サーバ装置は、受信した符号化ストリームを、参加メンバそれぞれのコンピュータ装置にネットワークを介して送信する。この符号化ストリームは、参加メンバそれぞれのコンピュータ装置に受信され、コンピュータ装置内のビデオ信号復号部624でこの発明に係る復号処理がなされる。ビデオ信号復号部624から出力された画像データが、コンピュータ装置の表示部に映像として表示される。

10

20

30

40

50



## 【0509】

すなわち、各参加メンバのコンピュータ装置の表示部には、他の参加メンバのビデオカメラ装置で撮影されたそれぞれの映像が表示されることになる。この発明の実施の第8の形態によれば、ビデオカメラ装置での撮影によるビデオ信号の符号化から、他の参加メンバのコンピュータ装置で復号されるまでの遅延時間が短く、参加メンバのコンピュータ装置の表示部に表示される他の参加メンバそれぞれの映像の違和感を少なくできる。

## 【0510】

さらに、ビデオ信号符号化部610をビデオカメラ装置側に搭載することも考えられる。例えば、ビデオカメラ装置に送信ユニット600を内蔵させる。このように構成することで、ビデオカメラ装置にコンピュータ装置などの他の装置を接続する必要がなくなる。

10

## 【0511】

このような、送信ユニット600が内蔵されたビデオカメラ装置と受信装置601とからなるシステムは、上述したテレビジョン会議システムの他にも、様々な応用が考えられる。例えば、図39に概略的に示されるように、このシステムを家庭用ゲーム機器に適用することができる。図39において、ビデオカメラ装置700は、この発明の実施の第8の形態による送信ユニット600が内蔵される。

## 【0512】

家庭用ゲーム機器の本体701は、例えばCPU、RAMおよびROMや、CD-ROM(Compact Disc-Read Only Memory)やDVD-ROM(Digital Versatile Disc-ROM)に対応したディスクドライブ装置、CPUにより生成された表示制御信号をビデオ信号に変換して出力するグラフィック制御部、オーディオ信号を再生するオーディオ再生部などが例えばバスで接続され、コンピュータ装置と略同様の構成とされる。家庭用ゲーム機器の本体701は、ROMに予め記憶されたプログラムや、ディスクドライブ装置に装填されたCD-ROMやDVD-ROMに記録されたプログラムに従い、CPUにより全体が制御される。RAMは、CPUのワークメモリとして用いられる。この家庭用ゲーム機器の本体701に対して、受信装置601を内蔵する。受信装置601から出力されるデジタルビデオ信号やその他の信号は、例えばバスを介してCPUに供給される。

20

## 【0513】

このようなシステムにおいて、例えば、家庭用ゲーム機器の本体において、外部から供給されたデジタルビデオ信号による画像を、ゲーム内の画像として用いることができるようにされたゲームソフトウェアが起動されているものとする。例えば、このゲームソフトウェアは、外部から供給されたデジタルビデオ信号による画像をゲーム内の画像として用いることができると共に、当該画像内での人物(プレイヤー)などの動きを識別し、識別された動きに応じた動作を行うことが可能とされている。

30

## 【0514】

ビデオカメラ装置700は、撮影されたデジタルビデオ信号を、内蔵される送信ユニット600において、ビデオ信号符号化部610でこの発明に係る符号化方法で符号化し、符号化ストリームをデジタル変調部611で変調して無線モジュール部612に供給し、アンテナ613から送信する。送信された電波は、家庭用ゲーム機器の本体701に内蔵される受信装置601においてアンテナ620で受信され、受信信号が無線モジュール部621およびフロントエンド部622を介してデジタル復調部623に供給される。受信信号は、デジタル復調部623で復調された符号化ストリームとされ、ビデオ信号復号部624に供給される。ビデオ信号復号部624では、供給された符号化ストリームを、この発明に係る復号方法で復号し、ベースバンドのデジタルビデオ信号を出力する。

40

## 【0515】

ビデオ信号復号部624から出力されたベースバンドのデジタルビデオ信号は、家庭用ゲーム機器の本体701において、バスに送出され、例えばRAMに一時的に記憶される。CPUは、所定のプログラムに従い、RAMに記憶されたデジタルビデオ信号を読み出すことで、このデジタルビデオ信号による画像内の人物の動きを検出したり、当該画像をゲーム内で用いたりすることができるようにされる。

50

## 【0516】

ビデオカメラ装置700で撮影され、得られたデジタルビデオ信号が符号化されてから、家庭用ゲーム機器の本体701で符号化ストリームが復号され画像が得られるまでの遅延時間が短いので、家庭用ゲーム機器の本体701上で動作するゲームソフトウェアにおける、プレーヤの動きに対する応答性が良くなり、ゲームの操作性を向上させることができる。

## 【0517】

なお、このような、家庭用ゲーム機器と共に用いられるビデオカメラ装置700は、価格や大きさなどの面からみても簡易な構成とされることが多く、コンピュータ装置などのように、処理能力の高いCPUや、記憶容量の大きなメモリを搭載できないことが想定される。

10

## 【0518】

つまり、一般的に、ビデオカメラ装置700は、ビデオカメラ装置700を利用するゲームをプレイする場合にのみ必要になる、家庭用ゲーム機器の本体701の周辺機器であり、家庭用ゲーム機器の本体701を用いてゲームをプレイするのに必要な装置ではない。そのような場合、ビデオカメラ装置700が家庭用ゲーム機器の本体701と別商品として販売される（所謂、別売り）ことが多い。その場合に、ビデオカメラ装置700に、処理能力の高いCPUや記憶容量の大きなメモリを搭載し、高額な価格で販売するようにすると、一般的に販売数が低減する恐れがある。その場合、このビデオカメラ装置700を利用するゲームの販売数も低減する恐れがあり、収益の低下につながる恐れがある。また、特に、家庭用ゲームにおいては普及率が販売数に対して強い影響を与えることも多く、ビデオカメラ装置700の普及率が低いと、さらに販売数が低減する恐れがある。

20

## 【0519】

逆に、ビデオカメラ装置700を安価で多数販売して普及率を向上させることにより、このビデオカメラ装置700を利用する家庭用ゲームの販売数や人気度を向上させることができ、さらに、そのことが家庭用ゲーム機器の本体701のさらなる購買動機に繋がることを期待することができる。そのためにも、ビデオカメラ装置700は、簡易な構成とすることが望ましい場合が多い。

## 【0520】

この場合、例えば、ビデオカメラ装置700に内蔵される送信ユニット600のビデオ信号符号化部610において、ウェーブレット変換を低い分解レベルで抑えて行うことが考えられる。こうすることで、係数並び替え用バッファ部に用いるメモリ容量が少なくて済む。

30

## 【0521】

また、ビデオ信号符号化部610に、実施の第3の形態で説明した、図30に例示される画像符号化装置の構成を適用することが考えられる。さらに、ビデオ信号符号化部610に、図33に例示される画像符号化装置の構成を適用すると、ビデオ信号符号化部610側でウェーブレット変換係数データの並び替え処理を行う必要が無くなるため、ビデオカメラ装置700側の負担をより減らすことが可能となり、好ましい。この場合、家庭用ゲーム機器の本体701側に内蔵される受信装置601において、ビデオ信号復号部624として、図34に例示される画像復号装置を用いる必要がある。

40

## 【0522】

なお、上述では、ビデオカメラ装置700と家庭用ゲーム機器の本体701とは、無線通信で接続されるように説明したが、これはこの例に限られない。すなわち、ビデオカメラ装置700と家庭用ゲーム機器の本体701とは、USBやIEEE1394といったインタフェースにより、有線で接続されていてもよい。

## 【0523】

以上のように、本発明の符号化方法（または復号方法）は、処理が容易であり、負荷や処理時間が少ないので、多様な形態に適用することができ、容易に多様な用途に応用することができる（すなわち汎用性が高い）ことも大きな効果である。

## 【0524】

50

上述した一連の処理は、ハードウェアにより実行させることもできるし、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータ、または、複数の装置よりなる情報処理システムの情報処理装置などに、プログラム記録媒体からインストールされる。

【0525】

図40は、上述した一連の処理をプログラムにより実行する情報処理システムの構成の例を示すブロック図である。

【0526】

図40に示されるように、情報処理システム800は、情報処理装置801、その情報処理装置801とPCIバス802によって接続された、記憶装置803、複数台のビデオテープレコーダ（VTR）であるVTR804-1乃至VTR804-S、ユーザがこれらに対する操作入力を行うためのマウス805、キーボード806、並びに操作コントローラ807により構成されるシステムであり、インストールされたプログラムによって、上述したような画像符号化処理や画像復号処理等を行うシステムである。

【0527】

例えば情報処理システム800の情報処理装置801は、RAID（Redundant Arrays of Independent Disks）でなる大容量の記憶装置803に記憶されている動画コンテンツを符号化して得られた符号化データを記憶装置803に記憶させたり、記憶装置803に記憶されている符号化データを復号して得られた復号画像データ（動画コンテンツ）を記憶装置803に記憶させたり、符号化データや復号画像データをVTR804-1乃至VTR804-Sを介してビデオテープに記録したりすることができる。また、情報処理装置801は、VTR804-1乃至VTR804-Sに装着されたビデオテープに記録された動画コンテンツを記憶装置803に取り込み得るようにもなされている。その際、情報処理装置801が、動画コンテンツを符号化するようにしてもよい。

【0528】

情報処理装置801は、マイクロプロセッサ901、GPU（Graphics Processing Unit）902、XDR（Extreme Data Rate）-RAM903、サウスブリッジ904、HDD905、USBインタフェース（USB I/F）906、およびサウンド入出力コーデック907を有している。

【0529】

GPU902は専用のバス911を介してマイクロプロセッサ901に接続される。XDR-RAM903は専用のバス912を介してマイクロプロセッサ901に接続される。サウスブリッジ904は、専用のバスを介してマイクロプロセッサ901のI/Oコントローラ944に接続される。このサウスブリッジ904には、HDD905、USBインタフェース906、および、サウンド入出力コーデック907も接続されている。このサウンド入出力コーデック907にはスピーカ921が接続されている。また、GPU902にはディスプレイ922が接続されている。

【0530】

またサウスブリッジ904には、さらに、PCIバス802を介して、マウス805キーボード806、VTR804-1乃至VTR804-S、記憶装置803、並びに、操作コントローラ807が接続されている。

【0531】

マウス805およびキーボード806は、ユーザの操作入力を受け、PCIバス802およびサウスブリッジ904を介して、ユーザの操作入力の内容を示す信号を、マイクロプロセッサ901に供給する。記憶装置803およびVTR804-1乃至VTR804-Sは、所定のデータを記録または再生できるようになされている。

【0532】

PCIバス802にはさらに、必要に応じてドライブ808が接続され、磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリなどのリムーバブルメディア811が適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じてHDD905にインス

10

20

30

40

50

トールされる。

【0533】

マイクロプロセッサ901は、OS (Operating System) 等の基本プログラムを実行する汎用のメインCPUコア941と、メインCPUコア941に内部バス945を介して接続された複数（この場合8個）のRISC (Reduced Instruction Set Computer) タイプの信号処理プロセッサである、サブCPUコア942-1乃至サブCPUコア942-8と、例えば256[MByte]の容量を持つXDR-RAM903に対するメモリコントロールを行うメモリコントローラ943と、サウスブリッジ904との間でデータの入出力を管理するI/O (In/Out) コントローラ944とが1チップに集積されたマルチコア構成でなり、例えば動作周波数4[GHz]を実現している。

【0534】

このマイクロプロセッサ901は、起動時、HDD905に格納された制御プログラムに基づき、HDD905に格納されている必要なアプリケーションプログラムを読み出してXDR-RAM903に展開し、この後このアプリケーションプログラム及びオペレータ操作に基づいて必要な制御処理を実行する。

【0535】

また、マイクロプロセッサ901は、ソフトウェアを実行することにより、例えば、上述した各実施の形態の画像符号化処理や画像復号処理を実現し、エンコードの結果得られた符号化ストリームを、サウスブリッジ904を介して、HDD905に供給して記憶させたり、デコードした結果得られる動画像コンテンツの再生映像を、GPU902へデータ転送して、ディスプレイ922に表示させたりすることができる。

【0536】

マイクロプロセッサ901内の各CPUコアの使用方法は任意であるが、例えば、メインCPUコア941が、画像符号化処理や画像復号処理の制御に関する処理を行い、8個のサブCPUコア942-1乃至サブCPUコア942-8に、ウェーブレット変換、係数並び替え、エントロピ符号化、エントロピ復号、ウェーブレット逆変換、量子化、および逆量子化等の各処理を、例えば図29を参照して説明したように同時並列的に実行させるようにしてもよい。その際、メインCPUコア941が、8個のサブCPUコア942-1乃至サブCPUコア942-8のそれぞれに対してラインブロック（プレシнкт）単位で処理を割り振るようにすれば、画像符号化処理や画像復号処理が、図29を参照して説明した場合と同様にラインブロック単位で同時並列的に実行される。つまり、画像符号化処理や画像復号処理の効率を向上させ、処理全体の遅延時間を短縮させ、さらに、負荷、処理時間、および、処理に必要なメモリ容量を低減させることができる。もちろん、これ以外の方法で各処理を行うようにしてもよい。

【0537】

例えば、マイクロプロセッサ901の8個のサブCPUコア942乃至サブCPUコア942-8のうちの一部がエンコード処理を、他の部分がデコード処理を、同時並列的に実行するようにすることも可能である。

【0538】

また、例えば、PCIバス802に、独立したエンコーダまたはデコーダ、もしくは、コーデック処理装置が接続されている場合、マイクロプロセッサ901の8個のサブCPUコア942乃至サブCPUコア942-8が、サウスブリッジ904およびPCIバス802を介して、これらの装置が実行する処理を制御するようにしてもよい。さらに、これらの装置が複数接続されている場合、または、これらの装置が複数のデコーダまたはエンコーダを含んでいる場合、マイクロプロセッサ901の8個のサブCPUコア942乃至サブCPUコア942-8は、複数のデコーダまたはエンコーダが実行する処理を、分担して制御するようにしてもよい。

【0539】

このときメインCPUコア941は、8個のサブCPUコア942乃至サブCPUコア942-8の動作を管理し、各サブCPUコアに対して処理を割り当てたり、処理結果を引き取ったりする。さらに、メインCPUコア941は、これらのサブCPUコアが行う以外の処理も行う。例えば、メインCPUコア941は、サウスブリッジ904を介してマウス805、キーボード806、または、操作コントローラ807から供給された命令を受け付け、命令に応じた種々の処理を実行する。

## 【0540】

GPU902は、ディスプレイ922に表示する動画コンテンツの再生映像を動かすときのテクスチャの張り込みなどに関する最終的なレンダリング処理に加えて、動画コンテンツの再生映像及び静止画コンテンツの静止画像をディスプレイ922に一度に複数表示するときの座標変換計算処理や、動画コンテンツの再生映像及び静止画コンテンツの静止画像に対する拡大・縮小処理等を行う機能を司り、マイクロプロセッサ901の処理負担を軽減させるようになされている。

## 【0541】

GPU902は、マイクロプロセッサ901の制御のもとに、供給された動画コンテンツの映像データや静止画コンテンツの画像データに対して所定の信号処理を施し、その結果得られた映像データや画像データをディスプレイ922へ送出して、画像信号をディスプレイ922へ表示させる。

10

## 【0542】

ところで、マイクロプロセッサ901における8個のサブCPUコア942乃至サブCPUコア942-8で同時並列的にデコードされた複数の動画コンテンツにおける再生映像は、バス911を介してGPU902へデータ転送されるが、このときの転送速度は、例えば、最大30[Gbyte/sec]であり、特殊効果の施された複雑な再生映像であっても高速かつ滑らかに表示し得るようになされている。

## 【0543】

一方、マイクロプロセッサ901は、動画コンテンツの映像データ及び音声データのうち音声データに対して音声ミキシング処理を施し、その結果得られた編集音声データを、サウブリッジ904およびサウンド入出力コーデック907を介して、スピーカ921へ送出することにより、音声信号に基づく音声をスピーカ921から出力させることもできる。

20

## 【0544】

上述した一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、ネットワークや記録媒体からインストールされる。

## 【0545】

この記録媒体は、例えば、図40に示されるように、装置本体とは別に、ユーザにプログラムを配信するために配布される、プログラムが記録されている磁気ディスク（フレキシブルディスクを含む）、光ディスク（CD-ROM、DVDを含む）、光磁気ディスク（MDを含む）、もしくは半導体メモリなどよりなるリムーバブルメディア811により構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに配信される、プログラムが記録されているHDD905や記憶装置803等で構成される。もちろん、記録媒体は、ROMやフラッシュメモリ等の半導体メモリであってもよい。

30

## 【0546】

以上においては、マイクロプロセッサ901内に8個のサブCPUコアが構成されるように説明したが、これに限らず、サブCPUコアの数は任意である。また、マイクロプロセッサ901が、メインCPUコアとサブCPUコアのような複数のコアにより構成されていなくてもよく、シングルコア（1つのコア）により構成されるCPUを用いるようにしてもよい。また、マイクロプロセッサ901の代わりに複数のCPUを用いるようにしてもよいし、複数の情報処理装置を用いる（すなわち、本発明の処理を実行するプログラムを、互いに連携して動作する複数の装置において実行する）ようにしてもよい。

40

## 【0547】

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

## 【0548】

また、本明細書において、システムとは、複数のデバイス（装置）により構成される装置全体を表わすものである。

## 【0549】

50

なお、以上において、一つの装置として説明した構成を分割し、複数の装置として構成するようにしてもよい。逆に、以上において複数の装置として説明した構成をまとめて一つの装置として構成されるようにしてもよい。また、各装置の構成に上述した以外の構成を付加するようにしてももちろんよい。さらに、システム全体としての構成や動作が実質的に同じであれば、ある装置の構成の一部を他の装置の構成に含めるようにしてもよい。

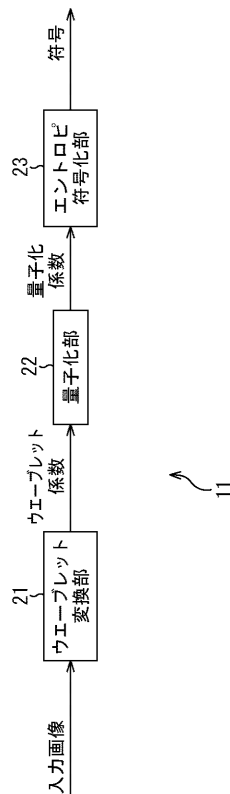
【符号の説明】

【 0 5 5 0 】

11 画像符号化装置, 23 エントロピ符号化部, 61 ライン判定部, 62 VLC符号化部, 63 最大有効桁数計算部, 64 VLC符号化部, 65 有効桁抽出部, 66 VLC符号化部, 67 サイン抽出部, 68 VLC符号化部, 111 画像復号装置, 151 符号分割部, 152 ライン判定部, 154 VLC復号部, 155 VLC復号部, 156 VLC復号部, 301 バッファ, 401 画像符号化装置, 410 ウェーブレット変換部, 411 途中計算用バッファ部, 412 係数並び替え用バッファ部, 413 係数並び替え部, 414 レート制御部, 415 エントロピ符号化部, 420 画像復号装置, 421 エントロピ復号部, 422 係数バッファ部, 423 ウェーブレット逆変換部, 430 画像符号化装置, 431 符号並び替え用バッファ部, 432 符号並び替え部, 441 画像符号化装置, 442 画像復号装置, 443 係数並び替え用バッファ部, 500 送信ユニット, 501 トライアックスケーブル, 502 カメラ制御部, 510 ビデオ信号符号化部, 511 ビデオ信号復号部, 526 ビデオ信号復号部, 527 ビデオ信号符号化部, 600 送信ユニット, 601 受信装置, 602 ビデオ信号符号化部, 612 無線モジュール部, 621 無線モジュール部, 624 ビデオ信号復号部, 700 ビデオカメラ装置, 701 家庭用ゲーム機器の本体

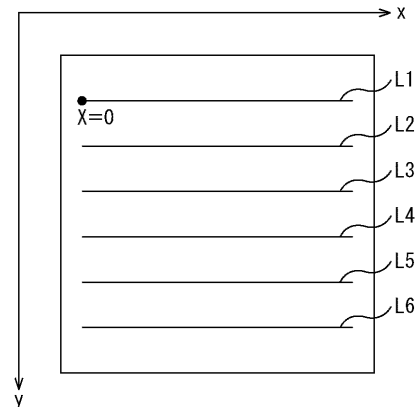
【図 1】

図1



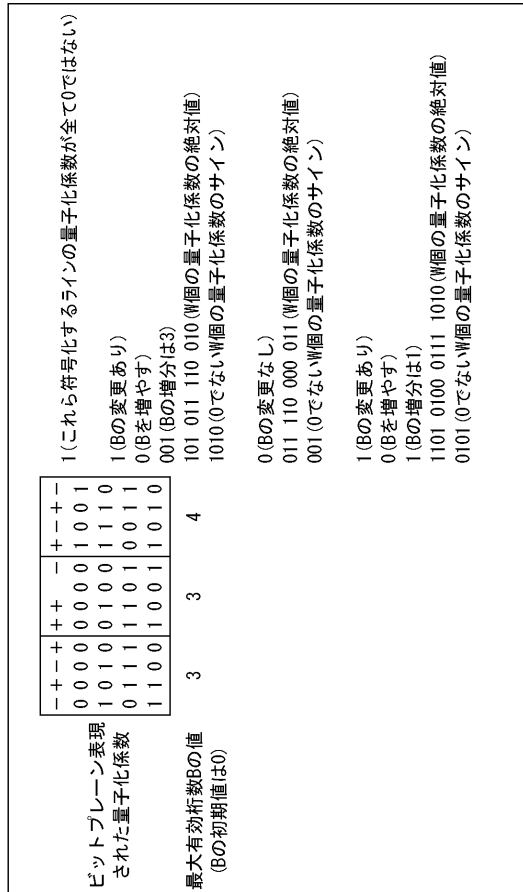
【図 2】

図2



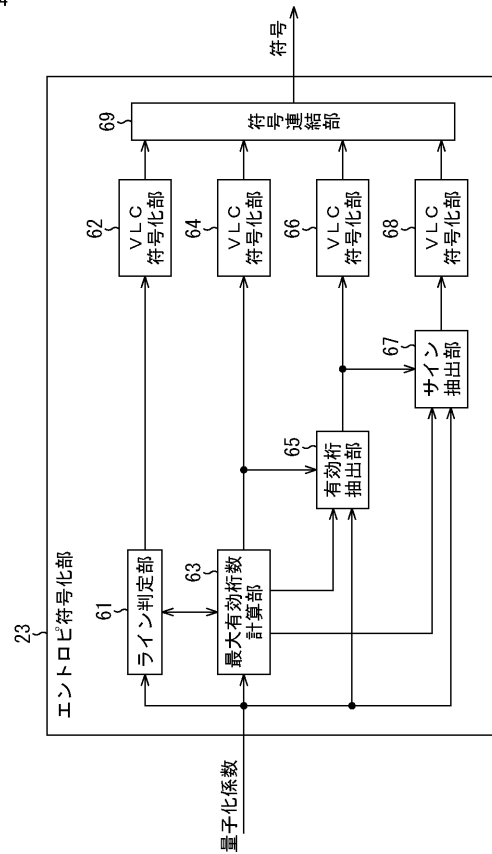
【 図 3 】

图3



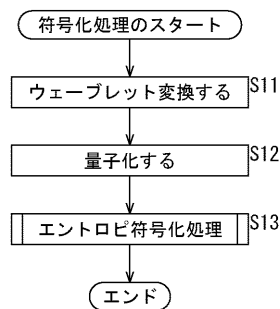
【 図 4 】

图4



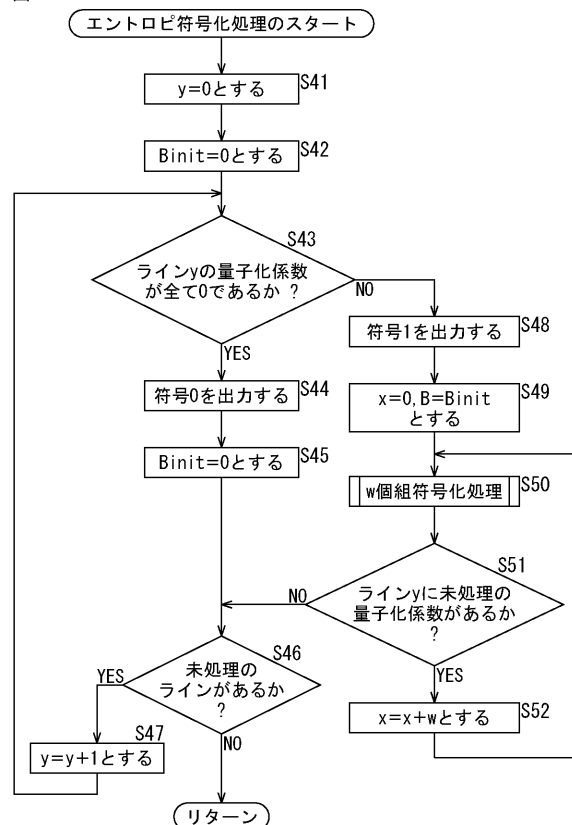
【 図 5 】

图5



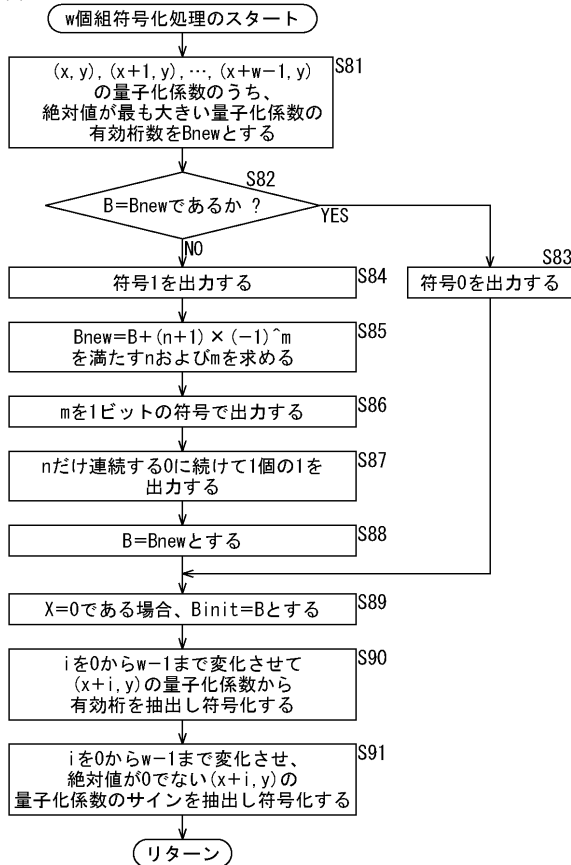
【 図 6 】

图6



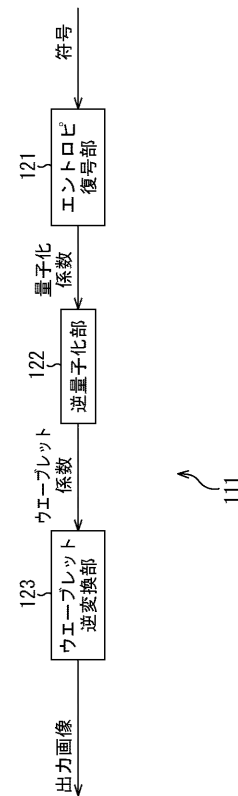
【図 7】

図7



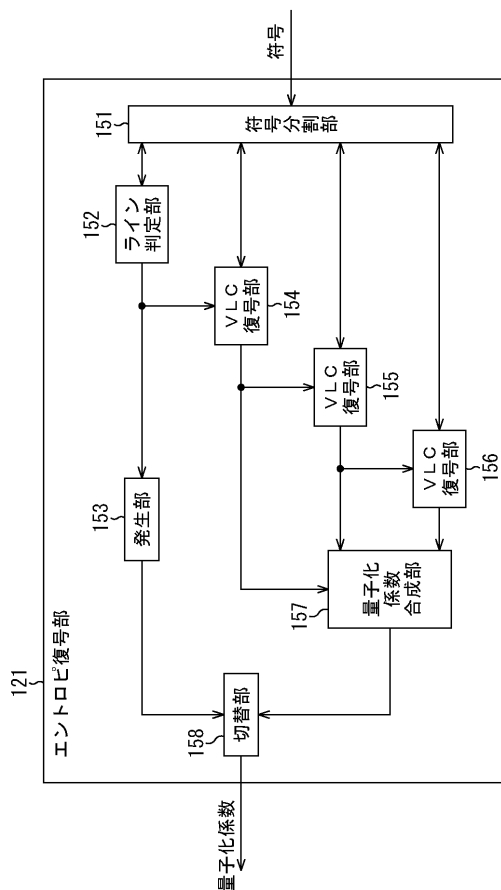
【図 8】

図8



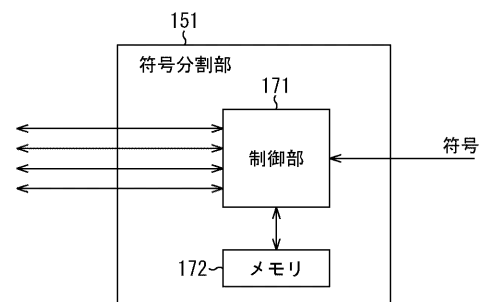
【図 9】

図9



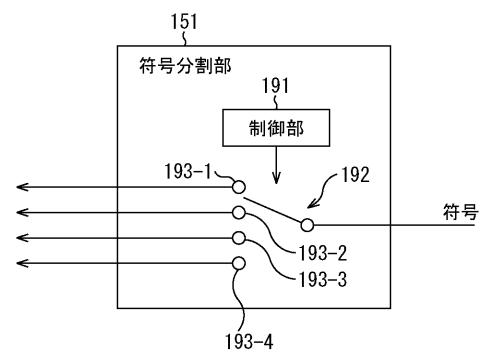
【図 10】

図10



【図 11】

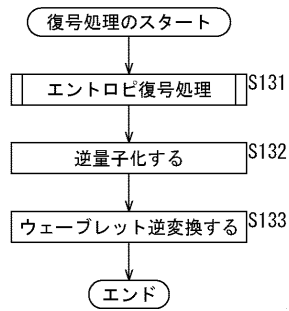
図11





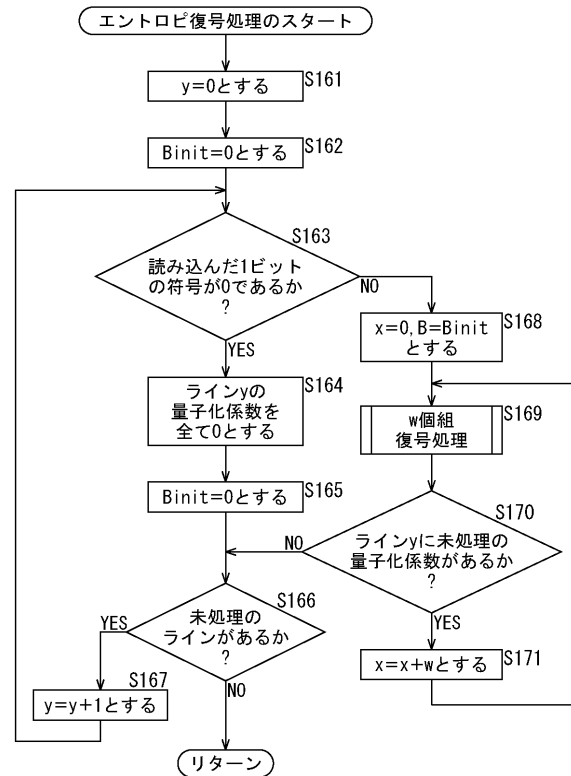
【図 1 2】

図12



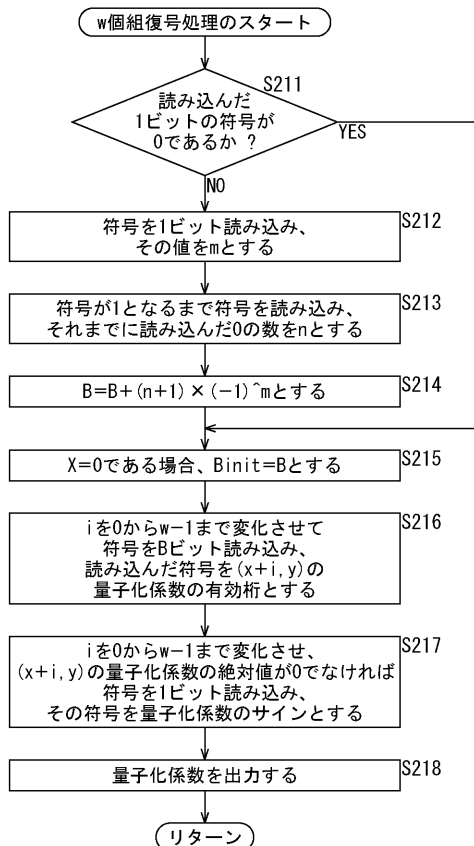
【図 1 3】

図13



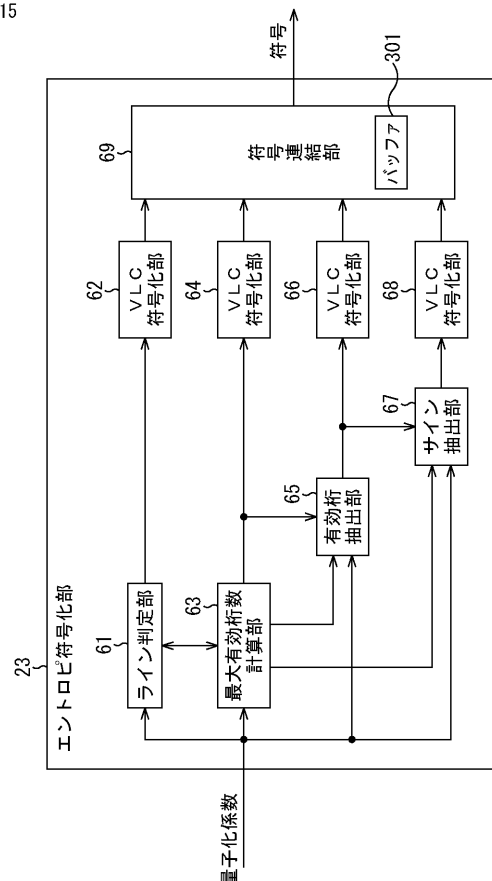
【図 1 4】

図14

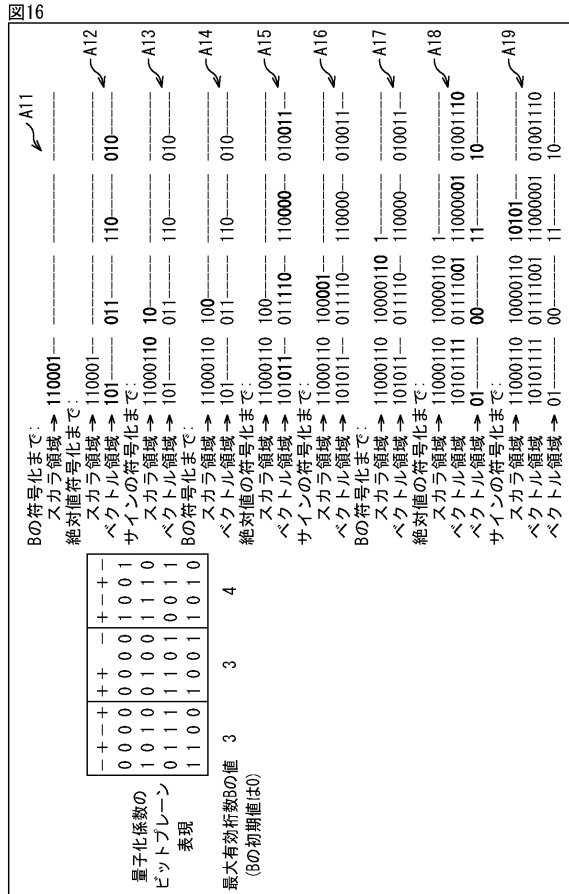


【図 1 5】

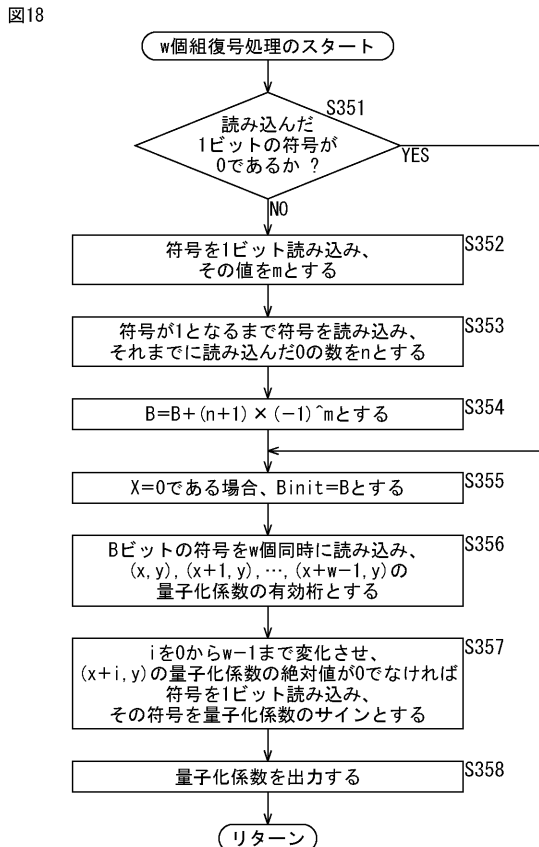
図15



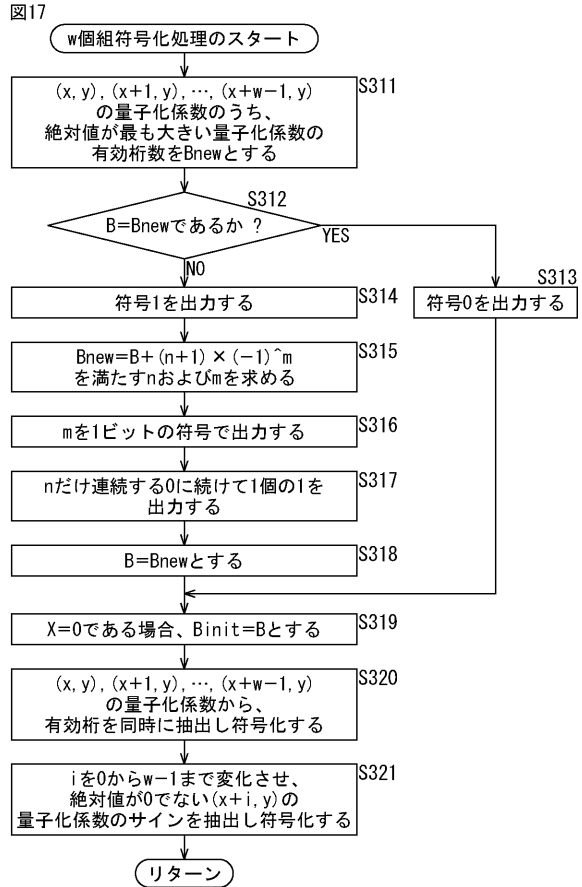
【図 16】



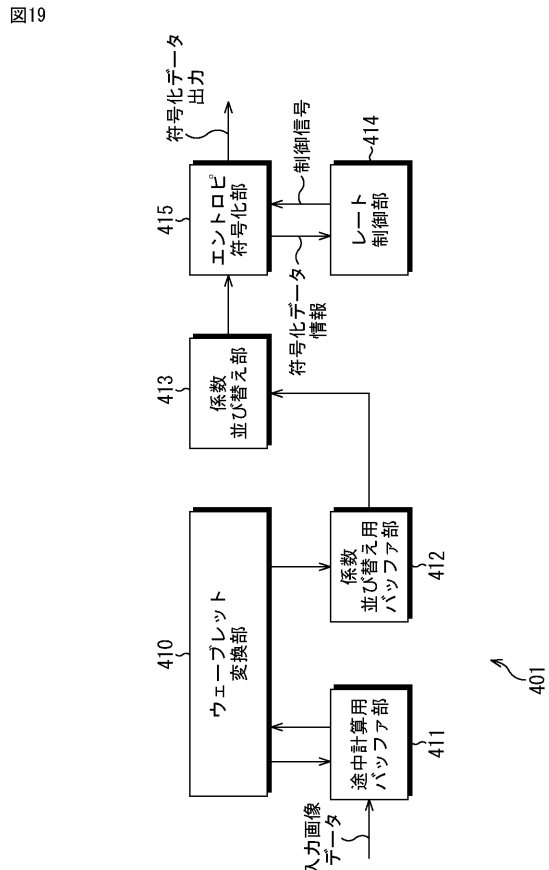
【図 18】



【図 17】

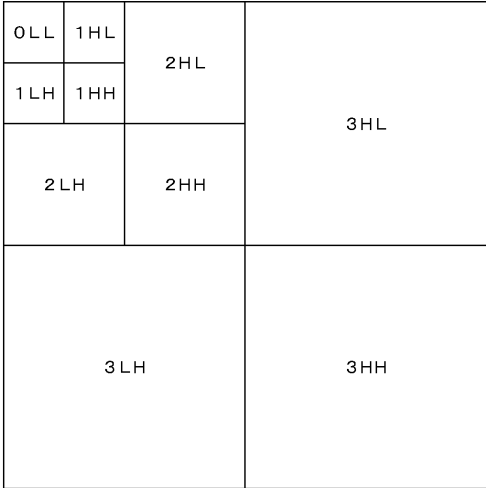


【図 19】



【図 20】

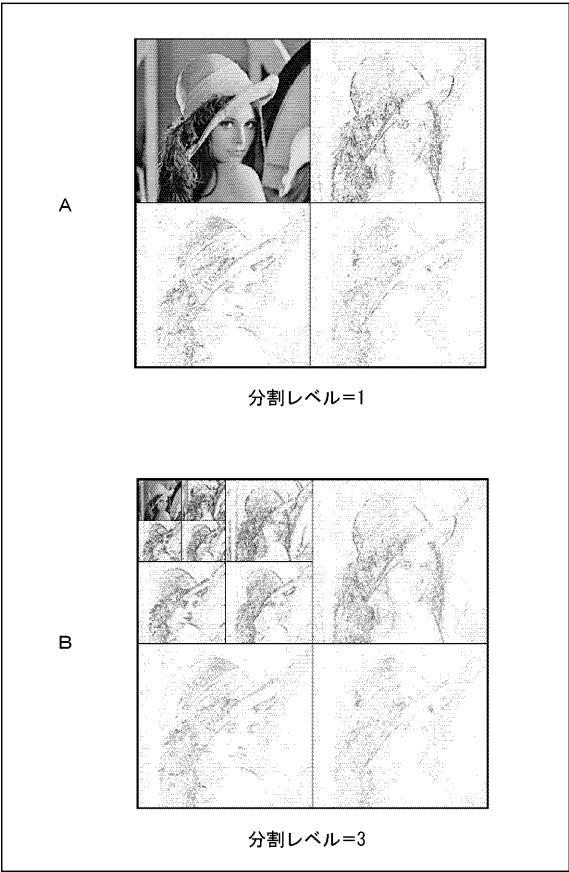
図20



分割レベル=3  
(H:高域、L:低域)

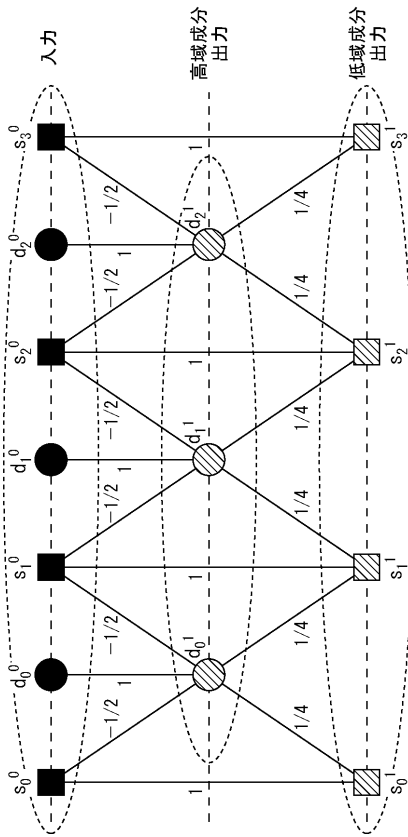
【図 21】

図21



【図 22】

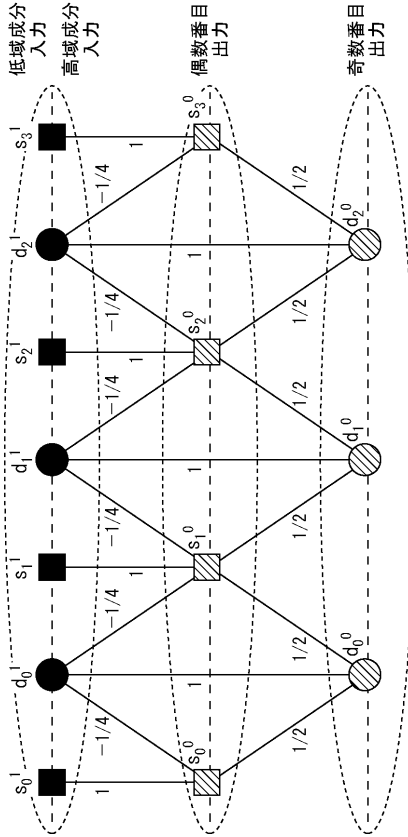
図22



$$d_i^1 = d_i^0 - 1/2(s_i^0 + s_{i+1}^0) : \text{高域成分出力}$$
$$s_i^1 = s_i^0 + 1/4(d_{i-1}^1 + d_i^1) : \text{低域成分出力}$$

【図 23】

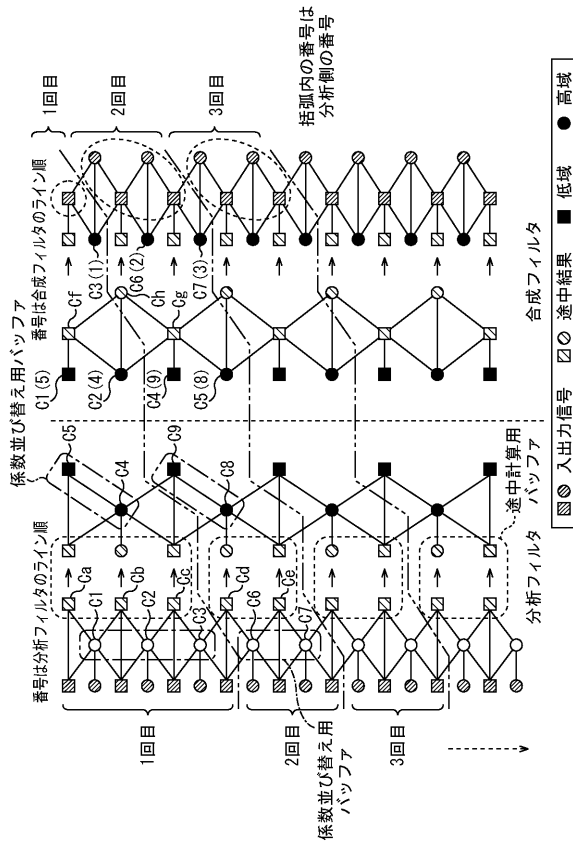
図23



$$s_i^0 = s_i^1 - 1/4(d_{i-1}^1 + d_i^1) : \text{偶数番目出力}$$
$$d_i^0 = d_i^1 + 1/2(s_i^0 + s_{i+1}^0) : \text{奇数番目出力}$$

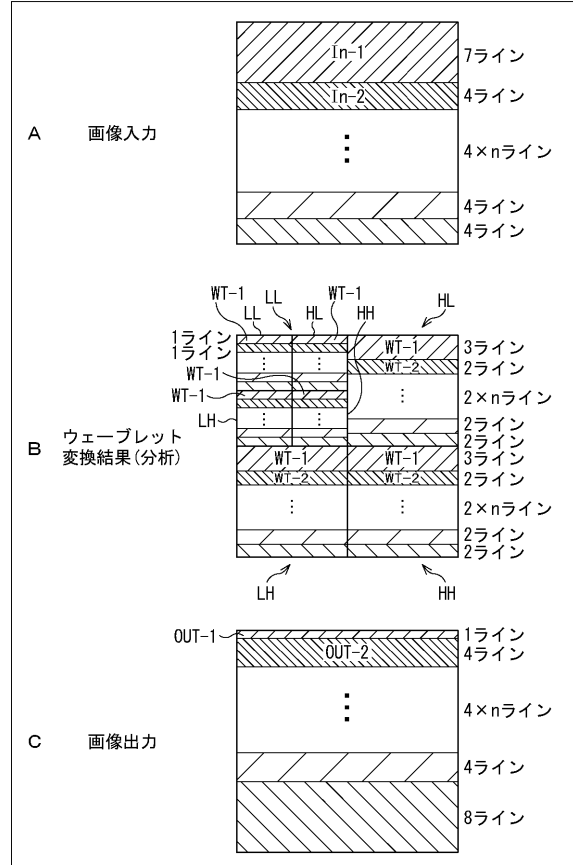
【図 24】

図24



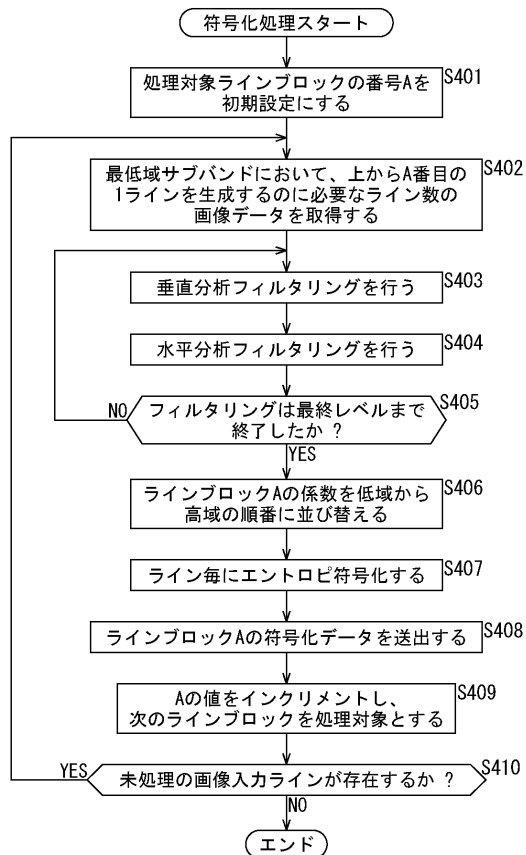
【図 25】

図25



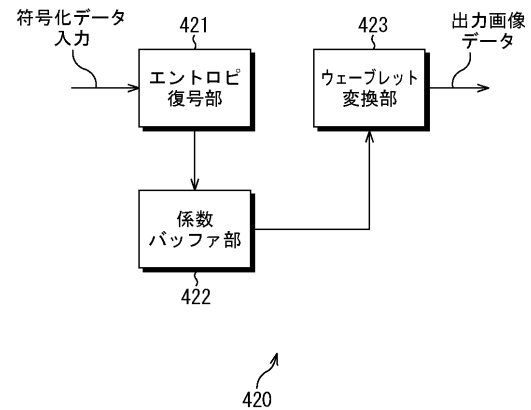
【図 26】

図26



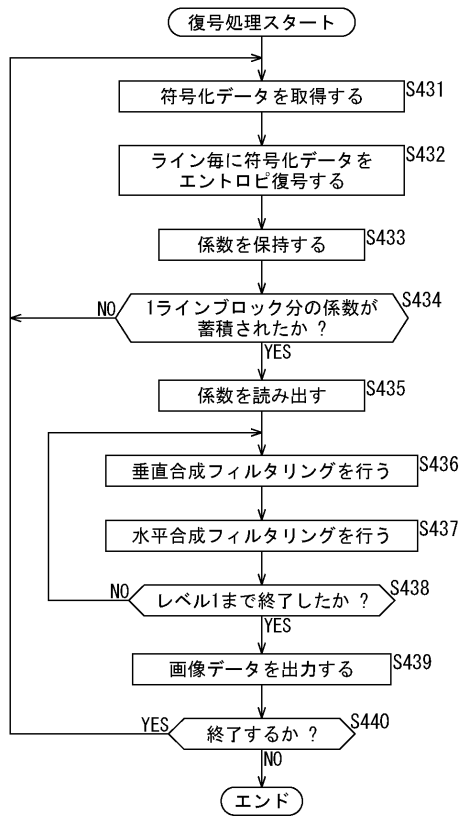
【図 27】

図27



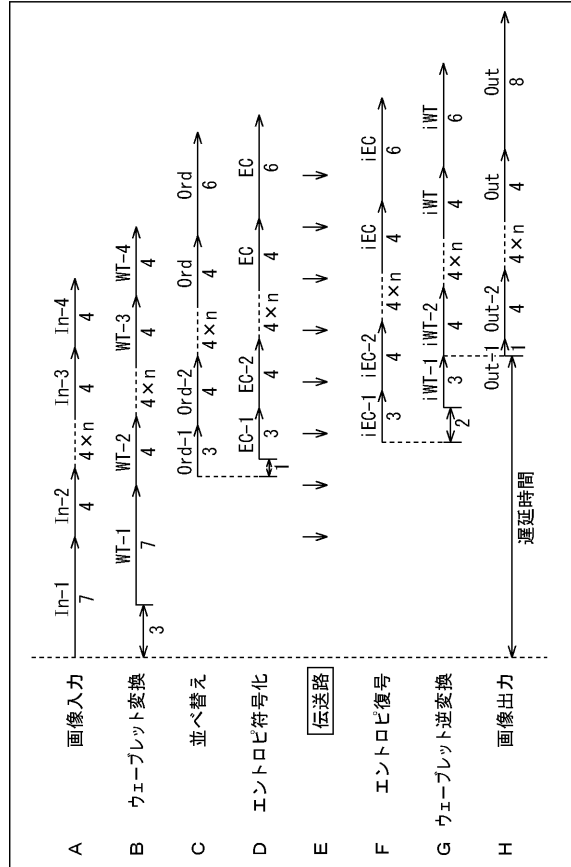
【図 28】

図28



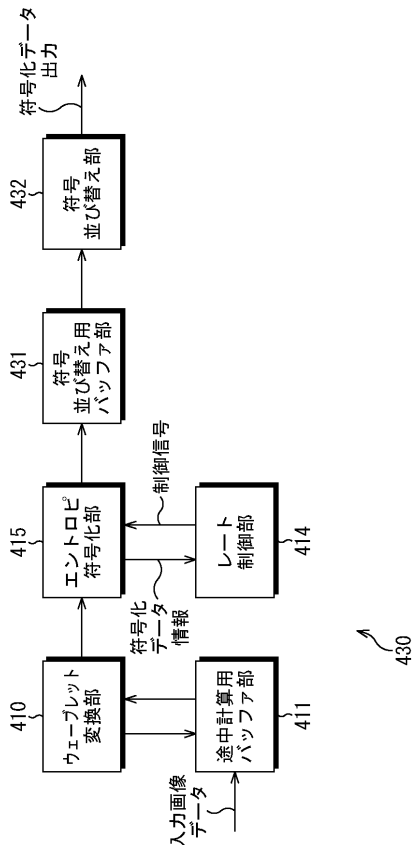
【図 29】

図29



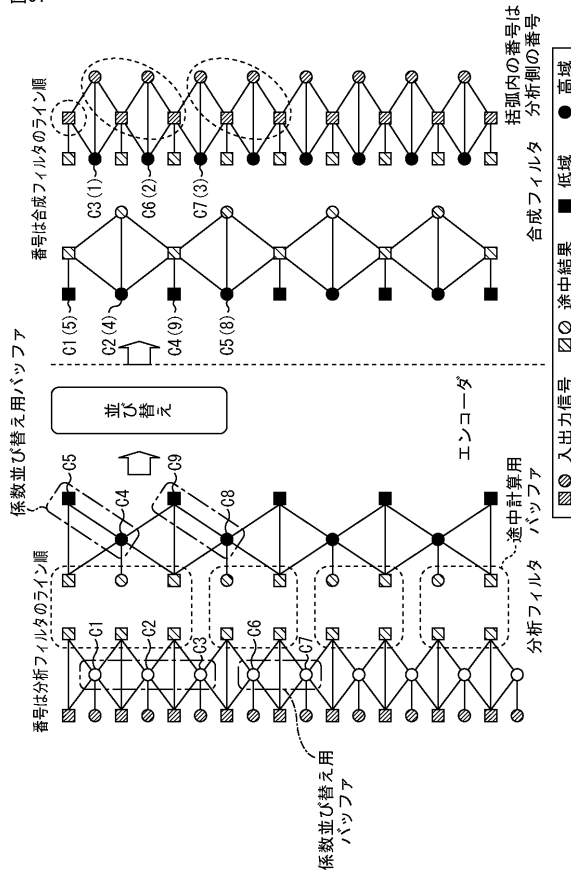
【図 30】

図30



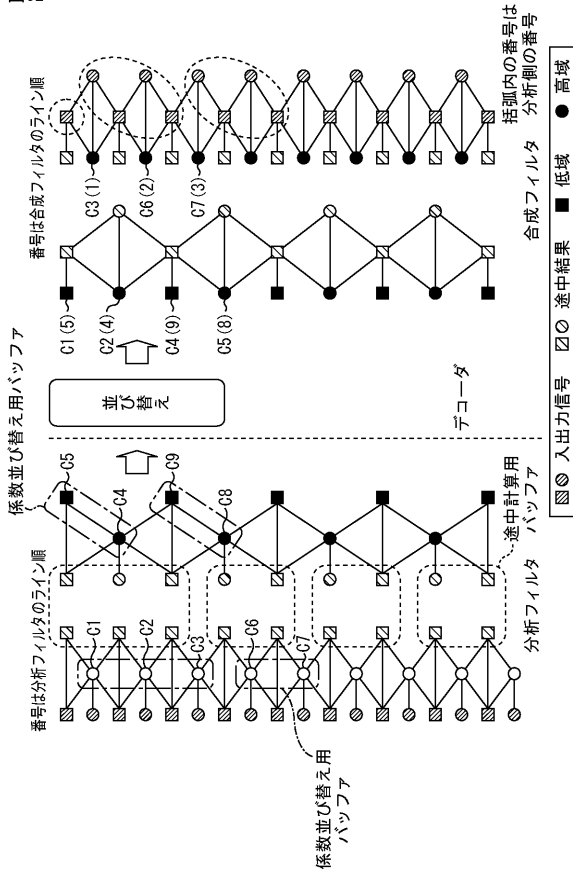
【図 31】

図31



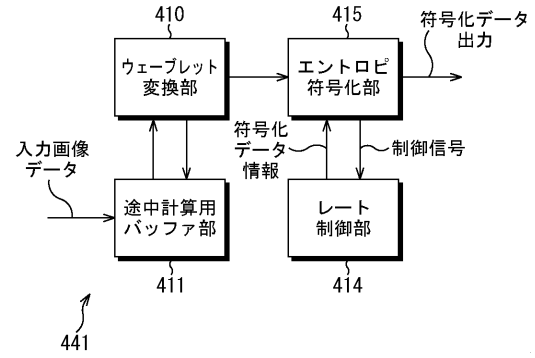
【図 3 2】

図32



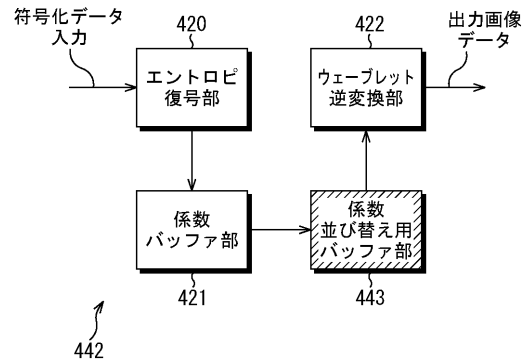
【図 3 3】

図33



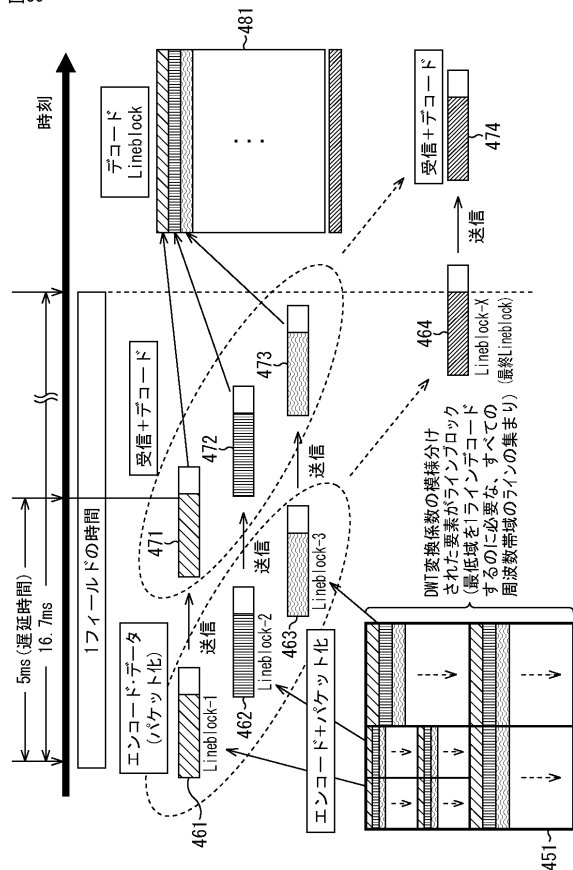
【図 3 4】

図34



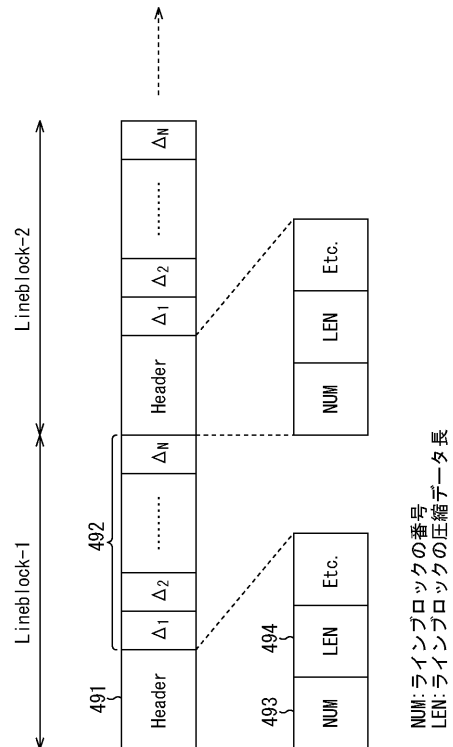
【図 3 5】

図35



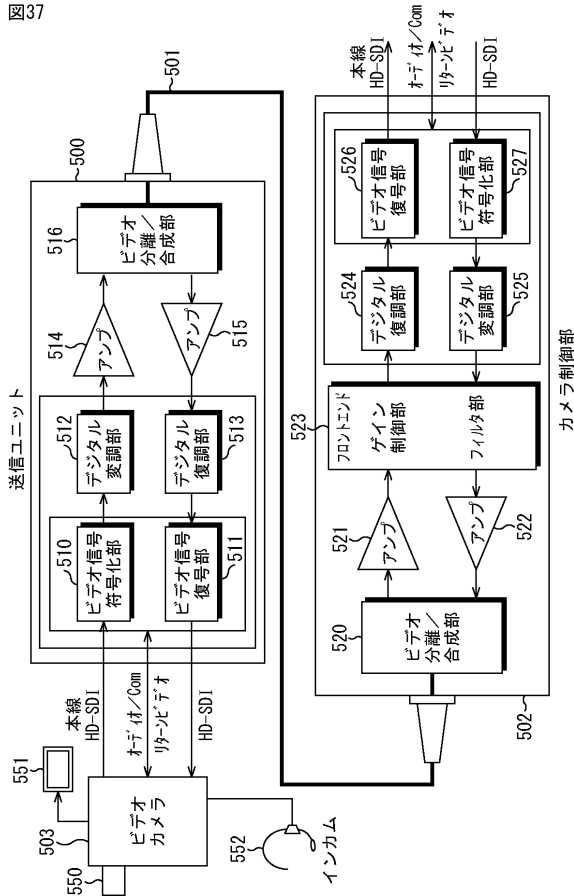
【図 3 6】

図36



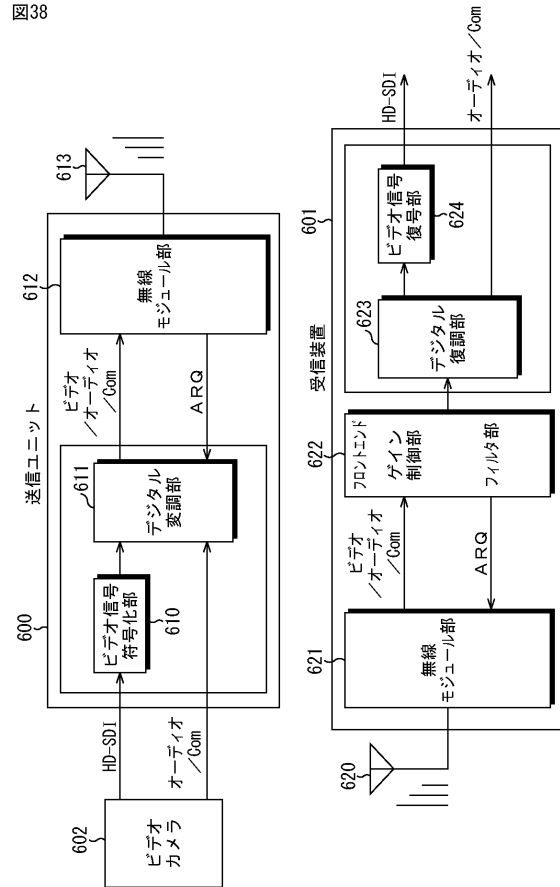
【 図 3 7 】

図37



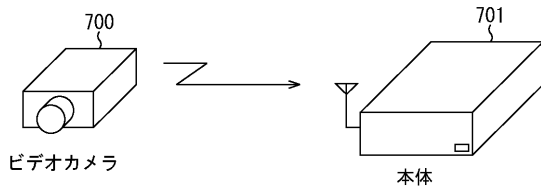
【 図 3 8 】

図38



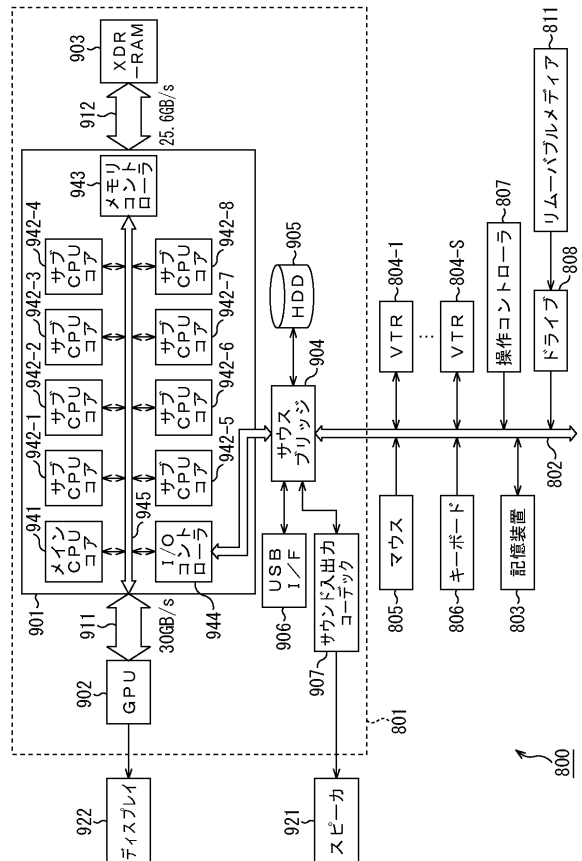
【 図 3 9 】

图39



【 図 4 0 】

図40



---

フロントページの続き

審査官 堀井 啓明

(56)参考文献 特開 2 0 0 3 - 2 0 4 4 3 9 ( J P , A )  
特開 2 0 0 1 - 1 9 7 5 0 0 ( J P , A )  
特開 2 0 0 5 - 1 0 9 7 7 6 ( J P , A )

(58)調査した分野(Int.Cl. , D B 名)  
H04N1/41-1/419  
H04N7/12-7/137